

# System SLIO

IM | 053-1CA00 | Manual

HB300 | IM | 053-1CA00 | en | 22-30

Interface module CANopen



YASKAWA Europe GmbH  
Philipp-Reis-Str. 6  
65795 Hattersheim  
Germany  
Tel.: +49 6196 569-300  
Fax: +49 6196 569-398  
Email: [info@yaskawa.eu](mailto:info@yaskawa.eu)  
Internet: [www.yaskawa.eu.com](http://www.yaskawa.eu.com)

## Table of contents

<b>1</b>	<b>General</b> .....	<b>5</b>
1.1	Copyright © YASKAWA Europe GmbH.....	5
1.2	About this manual.....	6
1.3	Safety information.....	6
<b>2</b>	<b>Basics and mounting</b> .....	<b>8</b>
2.1	Safety notes for the user.....	8
2.2	System conception.....	9
2.2.1	Overview.....	9
2.2.2	Components.....	10
2.2.3	Accessories.....	13
2.2.4	Hardware revision.....	15
2.3	Dimensions.....	15
2.4	Mounting bus coupler.....	18
2.5	Wiring.....	20
2.5.1	Wiring bus coupler.....	20
2.5.2	Wiring 8x periphery modules.....	23
2.5.3	Wiring 16x periphery modules.....	25
2.5.4	Wiring power modules.....	26
2.6	Demounting.....	30
2.6.1	Demounting bus coupler.....	30
2.6.2	Demounting 8x periphery modules.....	32
2.6.3	Demounting 16x periphery modules.....	35
2.7	Trouble shooting - LEDs.....	38
2.8	Industrial security and installation guidelines.....	39
2.8.1	Industrial security in information technology.....	39
2.8.2	Installation guidelines.....	41
2.9	General data for the System SLIO.....	43
2.9.1	Use in difficult operating conditions.....	45
<b>3</b>	<b>Hardware description</b> .....	<b>46</b>
3.1	Properties.....	46
3.2	Structure.....	47
3.2.1	Interfaces.....	47
3.2.2	Address selector.....	49
3.2.3	LEDs.....	49
3.3	Technical data.....	50
<b>4</b>	<b>Deployment</b> .....	<b>52</b>
4.1	Basics CAN.....	52
4.2	Fast introduction.....	54
4.3	Accessing the System SLIO.....	57
4.3.1	General.....	57
4.3.2	Accessing I/O area .....	57
4.3.3	Accessing parameter data .....	60
4.3.4	Accessing diagnostics data.....	61
4.4	Transfer rate and module-ID.....	62
4.5	LED status indication.....	64
4.6	Telegram structure.....	65
4.7	PDO .....	67

---

4.7.1	PDO overview.....	67
4.7.2	PDO Communication types.....	69
4.7.3	PDO transmission type.....	70
4.8	SDO.....	70
4.9	Object directory.....	73
4.10	Emergency Object.....	116
4.11	Network management.....	117
4.11.1	Overview.....	117
4.11.2	Node Guarding.....	118
4.11.3	Heartbeat.....	118

# 1 General

## 1.1 Copyright © YASKAWA Europe GmbH

### All Rights Reserved

This document contains proprietary information of Yaskawa and is not to be disclosed or used except in accordance with applicable agreements.

This material is protected by copyright laws. It may not be reproduced, distributed, or altered in any fashion by any entity (either internal or external to Yaskawa) except in accordance with applicable agreements, contracts or licensing, without the express written consent of Yaskawa and the business management owner of the material.

For permission to reproduce or distribute, please contact: YASKAWA Europe GmbH, European Headquarters, Philipp-Reis-Str. 6, 65795 Hattersheim, Germany

Tel.: +49 6196 569 300

Fax.: +49 6196 569 398

Email: [info@yaskawa.eu](mailto:info@yaskawa.eu)

Internet: [www.yaskawa.eu.com](http://www.yaskawa.eu.com)

### EC conformity declaration

Hereby, YASKAWA Europe GmbH declares that the products and systems are in compliance with the essential requirements and other relevant provisions. Conformity is indicated by the CE marking affixed to the product.

### Conformity Information

For more information regarding CE marking and Declaration of Conformity (DoC), please contact your local representative of YASKAWA Europe GmbH.

### Trademarks

SLIO and SPEED7 are registered trademarks of YASKAWA Europe GmbH.

CAN is a registered trademark of CAN in Automation e. V. (CiA).

All other trademarks, logos and service or product marks specified herein are owned by their respective companies.

### General terms of use

Every effort has been made to ensure that the information contained in this document was complete and accurate at the time of publishing. We cannot guarantee that the information is free of errors, and we reserve the right to change the information at any time. There is no obligation to inform the customer about any changes. The customer is requested to actively keep his documents up to date. The customer is always responsible for the deployment of the products with the associated documentation, taking into account the applicable directives and standards.

This documentation describes all hardware and software units and functions known today. It is possible that units are described that do not exist at the customer. The exact scope of delivery is described in the respective purchase contract.

### Document support

Contact your local representative of YASKAWA Europe GmbH if you have errors or questions regarding the content of this document. You can reach YASKAWA Europe GmbH via the following contact:

Email: [Documentation.HER@yaskawa.eu](mailto:Documentation.HER@yaskawa.eu)

### Technical support

Contact your local representative of YASKAWA Europe GmbH if you encounter problems or have questions regarding the product. If such a location is not available, you can reach the Yaskawa customer service via the following contact:

YASKAWA Europe GmbH,

European Headquarters, Philipp-Reis-Str. 6, 65795 Hattersheim, Germany

Tel.: +49 6196 569 500 (hotline)

Email: [support@yaskawa.eu](mailto:support@yaskawa.eu)

## 1.2 About this manual

### Objective and contents

This manual describes the IM 053CAN of the System SLIO.

- It describes the structure, configuration and application.
- The manual is targeted at users who have a background in automation technology.
- The manual consists of chapters. Each chapter describes a completed topic.
- For guidance, the manual provides:
  - An overall table of contents at the beginning of the manual.
  - References with pages numbers.

### Validity of the documentation

Product	Order no.	as of state:	
IM 053CAN	053-1CA00	HW: 01	FW: V1.0.0

### Icons Headings

Important passages in the text are highlighted by following icons and headings:



#### **DANGER!**

Immediate or likely danger. Personal injury is possible.



#### **CAUTION!**

Damages to property is likely if these warnings are not heeded.



*Supplementary information and useful tips.*

## 1.3 Safety information

### Applications conforming with specifications

The system is constructed and produced for:

- communication and process control
- general control and automation tasks
- industrial applications
- operation within the environmental conditions specified in the technical data
- installation into a cubicle



#### **DANGER!**

This device is not certified for applications in

- in explosive environments (EX-zone)

**Documentation**

The manual must be available to all personnel in the

- project design department
- installation department
- commissioning
- operation

**CAUTION!**

**The following conditions must be met before using or commissioning the components described in this manual:**

- Hardware modifications to the process control system should only be carried out when the system has been disconnected from power!
- Installation and hardware modifications only by properly trained personnel.
- The national rules and regulations of the respective country must be satisfied (installation, safety, EMC ...)

**Disposal**

**National rules and regulations apply to the disposal of the unit!**

## 2 Basics and mounting

### 2.1 Safety notes for the user



#### **DANGER!**

##### **Protection against dangerous voltages**

- When using System SLIO modules, the user must be protected from touching hazardous voltage.
- You must therefore create an insulation concept for your system that includes safe separation of the potential areas of ELV and hazardous voltage.
- Here, observe the insulation voltages between the potential areas specified for the System SLIO modules and take suitable measures, such as using PELV/SELV power supplies for System SLIO modules.

#### **Handling of electrostatic sensitive modules**

The modules are equipped with highly integrated components in MOS technology. These components are highly sensitive to over-voltages that occur, e.g. with electrostatic discharge. The following symbol is used to identify these hazardous modules:



The symbol is located on modules, module racks or on packaging and thus indicates electrostatic sensitive modules. Electrostatic sensitive modules can be destroyed by energies and voltages that are far below the limits of human perception. If a person who is not electrically discharged handles electrostatic sensitive modules, voltages can occur and damage components and thus impair the functionality of the modules or render the modules unusable. Modules damaged in this way are in most cases not immediately recognized as faulty. The error can only appear after a long period of operation. Components damaged by static discharge can show temporary faults when exposed to temperature changes, vibrations or load changes. Only the consistent use of protective devices and responsible observance of the handling rules can effectively prevent malfunctions and failures on electrostatic sensitive modules.

#### **Shipping of modules**

Please always use the original packaging for shipping.

#### **Measurement and modification of electrostatic sensitive modules**

For measurements on electrostatic sensitive modules the following must be observed:

- Floating measuring instruments must be discharged before use.
- Measuring instruments used must be grounded.

When modifying electrostatic sensitive modules, ensure that a grounded soldering iron is used.



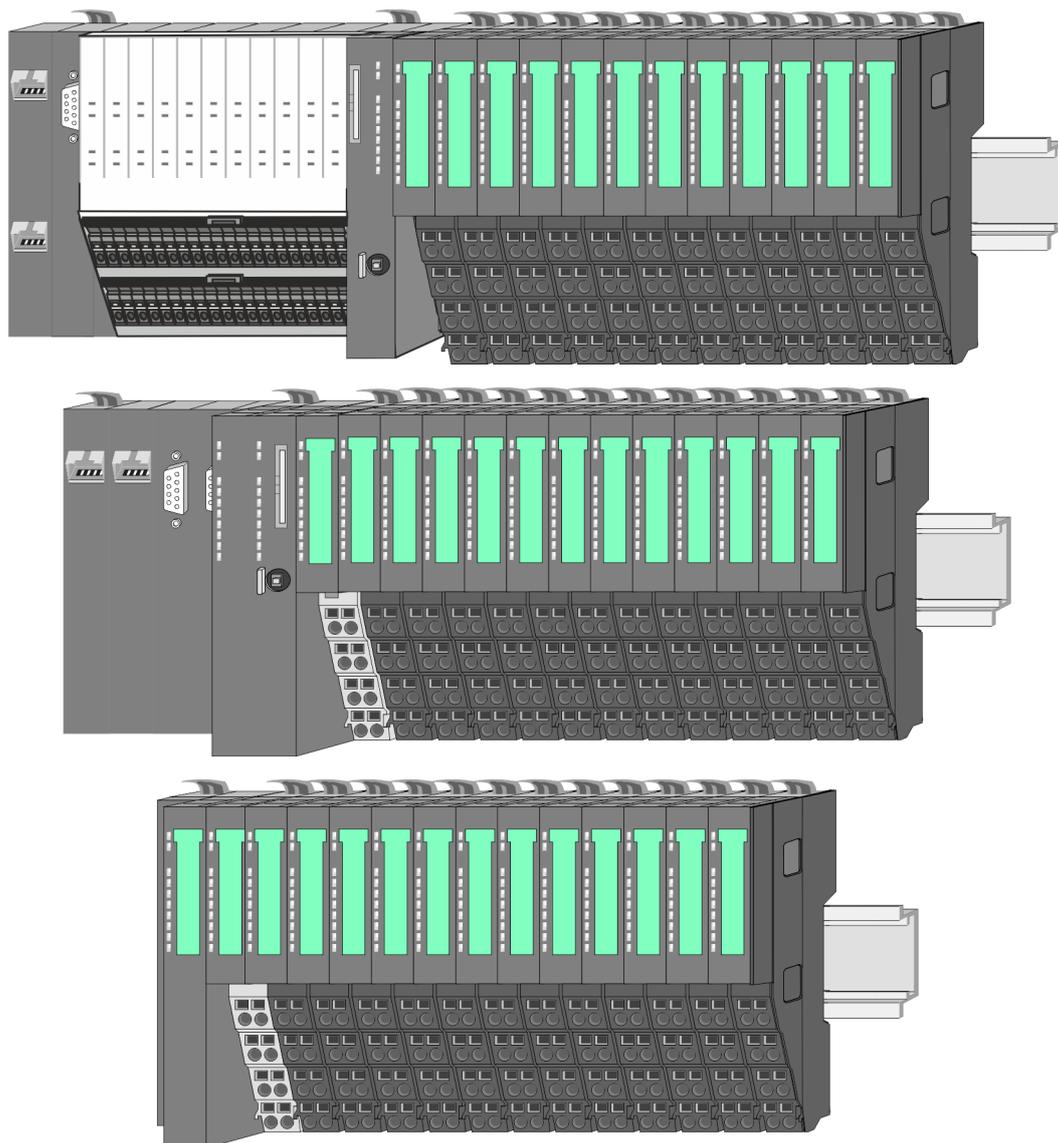
#### **CAUTION!**

When working with and on electrostatic sensitive modules, make sure that personnel and equipment are adequately grounded.

## 2.2 System conception

### 2.2.1 Overview

The System SLIO is a modular automation system for assembly on a 35mm mounting rail. By means of the periphery modules with 2, 4, 8 and 16 channels this system may properly be adapted matching to your automation tasks. The wiring complexity is low, because the supply of the DC 24V power section supply is integrated to the backplane bus and defective modules may be replaced with standing wiring. By deployment of the power modules in contrasting colors within the system, further isolated areas may be defined for the DC 24V power section supply, respectively the electronic power supply may be extended with 2A.



## 2.2.2 Components

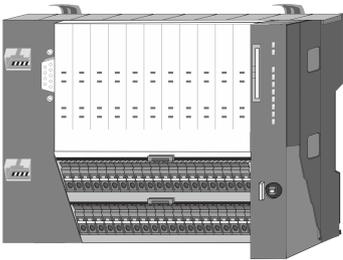
- CPU (head module)
- Bus coupler (head module)
- Line extension
- 8x periphery modules
- 16x periphery modules
- Power modules
- Accessories



**CAUTION!**

Only Yaskawa modules may be combined. A mixed operation with third-party modules is not allowed!

### CPU 01xC



With the CPU 01xC electronic, input/output components and power supply are integrated to one casing. In addition, up to 64 periphery modules of the System SLIO can be connected to the backplane bus. As head module via the integrated power module for power supply CPU electronic and the I/O components are supplied as well as the electronic of the periphery modules, which are connected via backplane bus. To connect the power supply of the I/O components and for DC 24V power section supply of via backplane bus connected periphery modules, the CPU has removable connectors. By installing of up to 64 periphery modules at the backplane bus, these are electrically connected, this means these are assigned to the backplane bus, the electronic modules are power supplied and each periphery module is connected to the DC 24V power section supply.

### CPU 01x



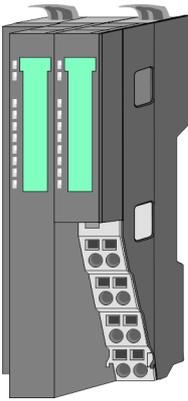
With this CPU 01x, CPU electronic and power supply are integrated to one casing. As head module, via the integrated power module for power supply, CPU electronic and the electronic of the connected periphery modules are supplied. The DC 24V power section supply for the linked periphery modules is established via a further connection of the power module. By installing of up to 64 periphery modules at the backplane bus, these are electrically connected, this means these are assigned to the backplane bus, the electronic modules are power supplied and each periphery module is connected to the DC 24V power section supply.



**CAUTION!**

CPU part and power module may not be separated!  
Here you may only exchange the electronic module!

### Bus coupler



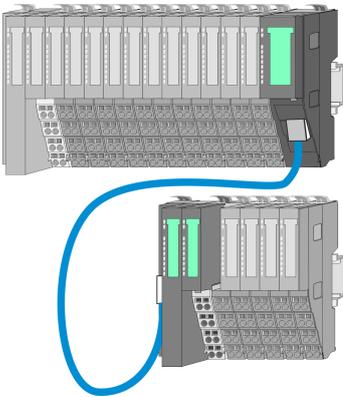
With a bus coupler bus interface and power module is integrated to one casing. With the bus interface you get access to a subordinated bus system. As head module, via the integrated power module for power supply, bus interface and the electronic of the connected periphery modules are supplied. The DC 24V power section supply for the linked periphery modules is established via a further connection of the power module. By installing of up to 64 periphery modules at the bus coupler, these are electrically connected, this means these are assigned to the backplane bus, the electronic modules are power supplied and each periphery module is connected to the DC 24V power section supply.



#### CAUTION!

Bus interface and power module may not be separated!  
Here you may only exchange the electronic module!

### Line extension

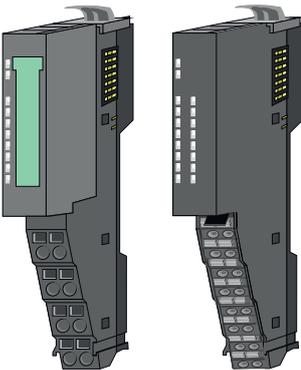


In the System SLIO there is the possibility to place up to 64 modules in on line. By means of the line extension you can divide this line into several lines. Here you have to place a line extension master at each end of a line and the subsequent line has to start with a line extension slave. Master and slave are to be connected via a special connecting cable. In this way, you can divide a line on up to 5 lines. For each line extension the maximum number of pluggable modules at the System SLIO bus is decreased by 1. To use the line extension no special configuration is required.



*Please note that some modules do not support line extensions due to the system. For more information, see the 'System SLIO - Compatibility List' at [www.yaskawa.eu.com](http://www.yaskawa.eu.com)*

### Periphery modules

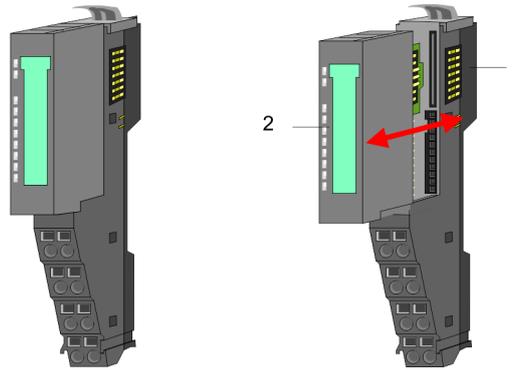


The periphery modules are available in the following 2 versions, whereby of each the electronic part can be replaced with standing wiring:

- 8x periphery module for a maximum of 8 channels.
- 16x periphery module for a maximum of 16 channels.

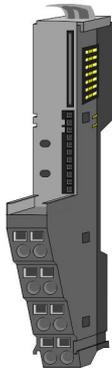
**8x periphery modules**

Each 8x periphery module consists of a *terminal* and an *electronic module*.



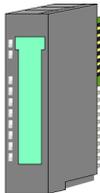
- 1 Terminal module
- 2 Electronic module

**Terminal module**



The *terminal* module serves to carry the electronic module, contains the backplane bus with power supply for the electronic, the DC 24V power section supply and the staircase-shaped terminal for wiring. Additionally the terminal module has a locking system for fixing at a mounting rail. By means of this locking system your system may be assembled outside of your switchgear cabinet to be later mounted there as whole system.

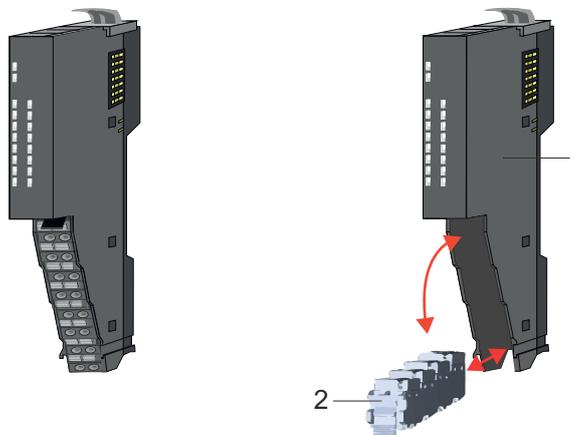
**Electronic module**



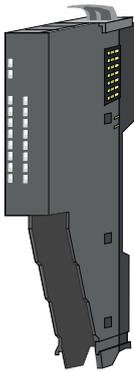
The functionality of a periphery module is defined by the *electronic module*, which is mounted to the terminal module by a sliding mechanism. With an error the defective electronic module may be exchanged for a functional module with standing installation. At the front side there are LEDs for status indication. For simple wiring each module shows corresponding connection information at the front and at the side.

**16x periphery modules**

Each 16x periphery module consists of an *electronic unit* and a *terminal block*.



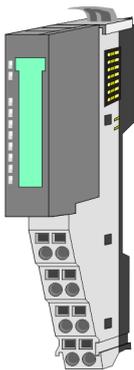
- 1 Electronic unit
- 2 Terminal block

**Electronic unit**

The functionality of a 16x periphery module is defined via the terminal block, which is connected to the *electronic unit* via a secure flap mechanism. In the case of an error you can exchange the defective electronic unit for a functional unit with standing wiring. At the front side there are LEDs for status indication. For easy wiring each electronic unit shows corresponding connection information at the side. The electronic unit provides the slot for the terminal block for the wiring and contains the backplane bus with power supply for the electronic and the connection to the DC 24V power section supply. Additionally the electronic unit has a locking system for fixing it at a mounting rail. By means of this locking system your system may be assembled outside of your switchgear cabinet to be later mounted there as whole system.

**Terminal block**

The *terminal block* provides the electrical interface for the signalling and supplies lines of the module. When mounting the terminal block, it is attached to the bottom of the electronic unit and turned towards the electronic unit until it clicks into place. With the wiring a "push-in" spring-clip technique is used. This allows a quick and easy connection of your signal and supply lines. The clamping off takes place by means of a screwdriver.

**Power module**

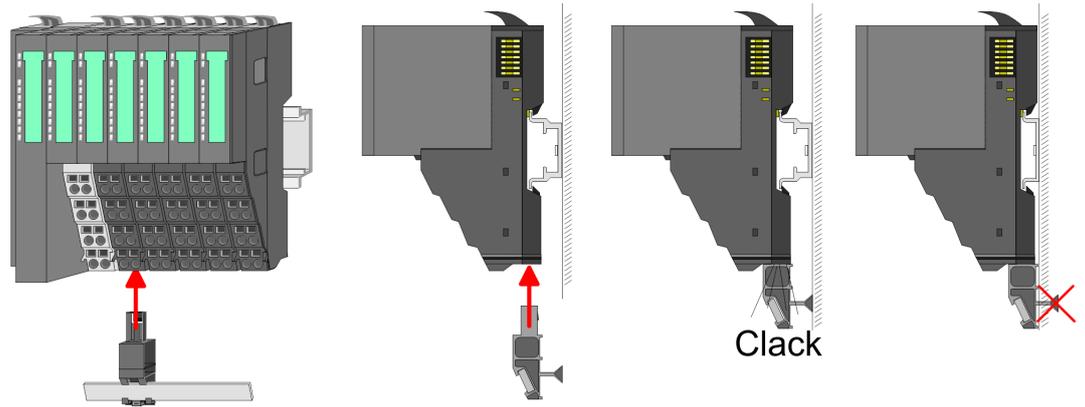
In the System SLIO the power supply is established by power modules. These are either integrated to the head module or may be installed between the periphery modules. Depending on the power module isolated areas of the DC 24V power section supply may be defined respectively the electronic power supply may be extended with 2A. For better recognition the colour of the power modules are contrasting to the periphery modules.

**2.2.3 Accessories****Shield bus carrier**

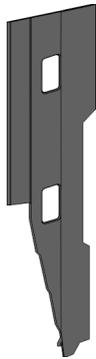
*Please note that a shield bus carrier cannot be mounted on a 16x periphery module!*



The shield bus carrier (order no.: 000-0AB00) serves to carry the shield bus (10mm x 3mm) to connect cable shields. Shield bus carriers, shield bus and shield fixings are not in the scope of delivery. They are only available as accessories. The shield bus carrier is mounted underneath the terminal of the terminal module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.



**Bus cover**



With each head module, to protect the backplane bus connectors, there is a mounted bus cover in the scope of delivery. You have to remove the bus cover of the head module before mounting a System SLIO module. For the protection of the backplane bus connector you always have to mount the bus cover at the last module of your system again. The bus cover has the order no. 000-0AA00.

**Coding pins**



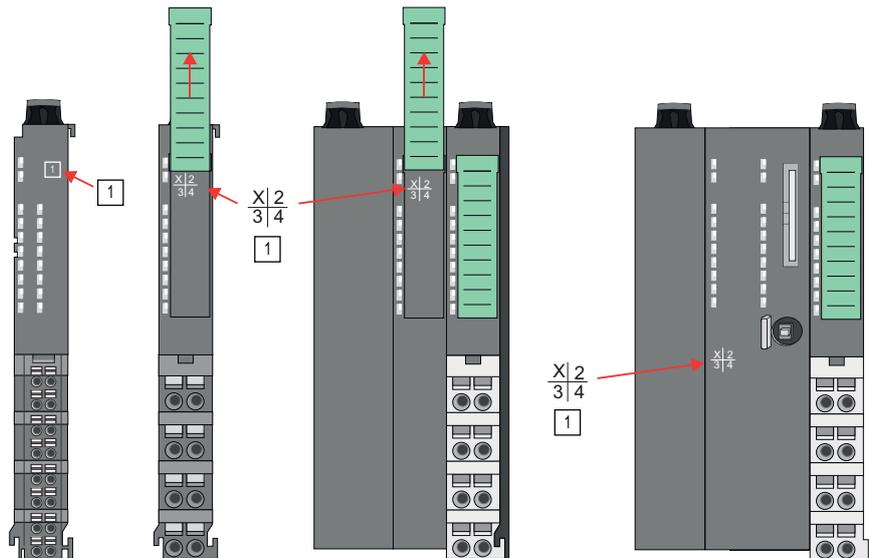
*Please note that a coding pin cannot be installed on a 16x periphery module! Here you have to make sure that the associated terminal block is plugged again when the electronics unit is replaced.*

There is the possibility to fix the assignment of electronic and terminal module. Here coding pins (order number 000-0AC00) can be used. The coding pin consists of a coding jack and a coding plug. By combining electronic and terminal module with coding pin, the coding jack remains in the electronic module and the coding plug in the terminal module. This ensures that after replacing the electronic module just another electronic module can be plugged with the same encoding.

### 2.2.4 Hardware revision

#### Hardware revision on the front

- The hardware revision is printed on every System SLIO module.
- Since a System SLIO 8x peripheral module consists of a terminal and electronic module, you will find a hardware revision printed on each of them.
- Authoritative for the hardware revision of a System SLIO module is the hardware revision of the electronic module. This is located under the labeling strip of the corresponding electronic module.
- Depending on the module type, there are the following 2 variants e.g. to indicate hardware revision 1:
  - Current modules have a 1 on the front.
  - With earlier modules, the 1 is marked with 'X' on a number grid.



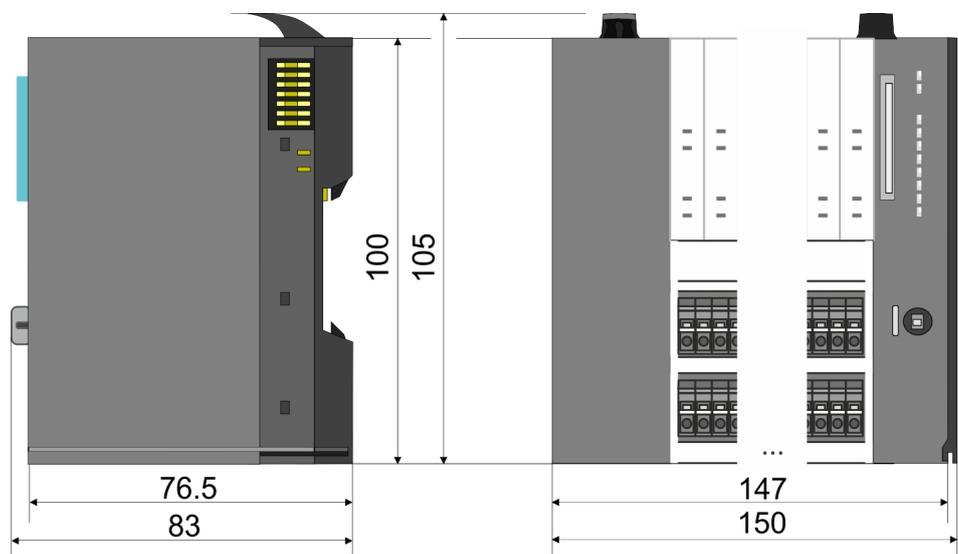
#### Hardware revision via web server

On the CPUs and some bus couplers, you can check the hardware revision 'HW Revision' via the integrated web server.

## 2.3 Dimensions

### CPU 01xC

All dimensions are in mm.

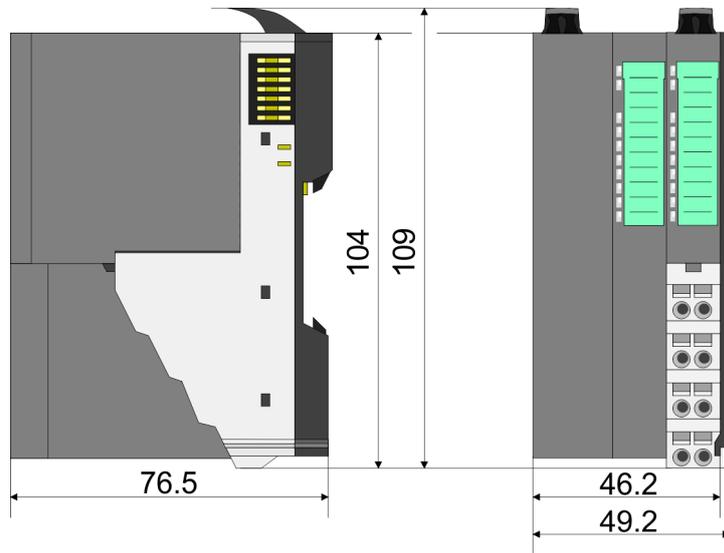


Dimensions

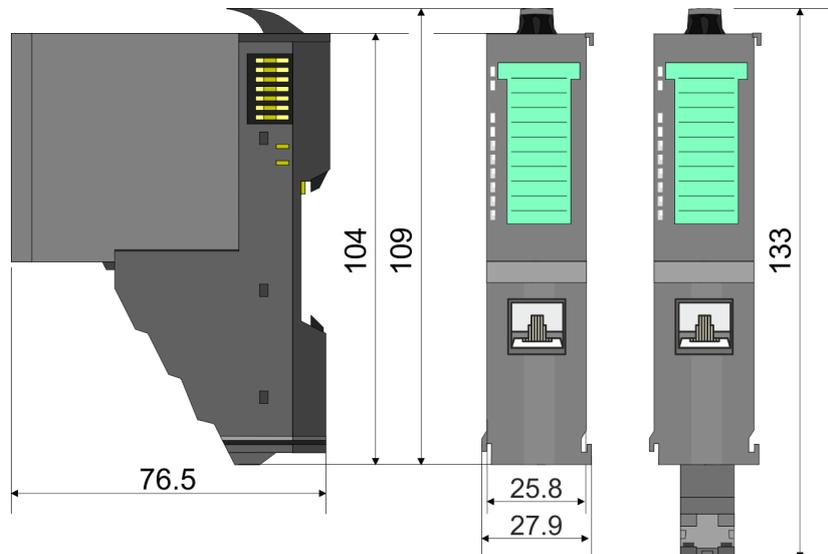
CPU 01x

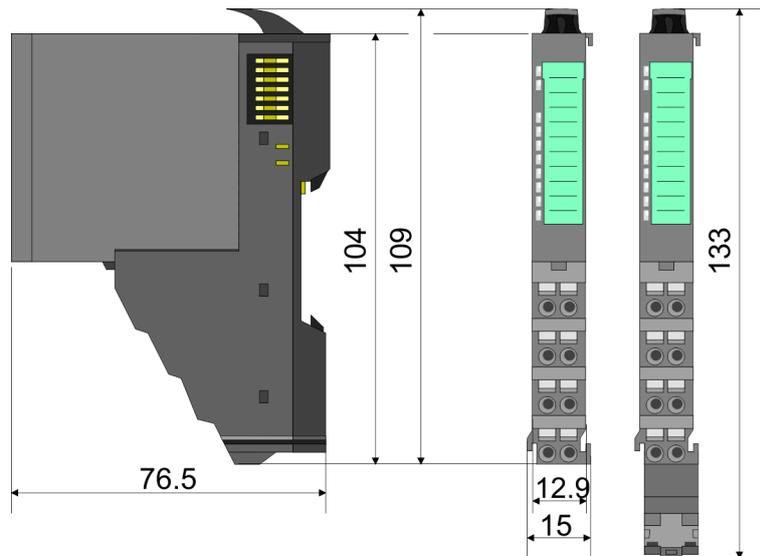
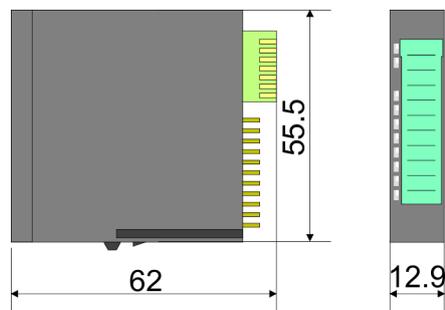
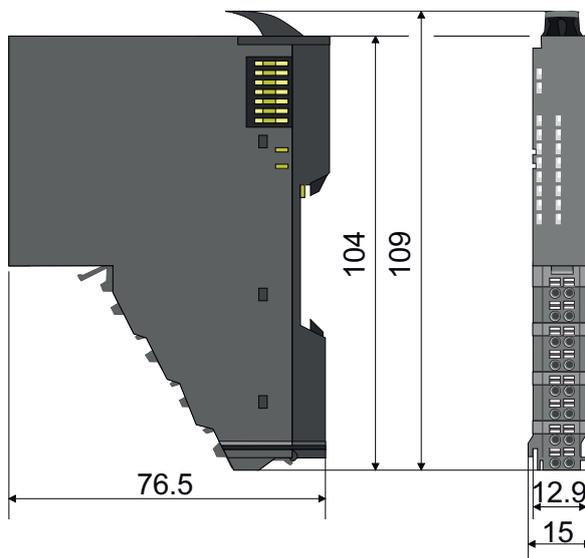


Bus coupler and line extension slave



Line extension master



**8x periphery module****Electronic module****16x periphery module**

## 2.4 Mounting bus coupler

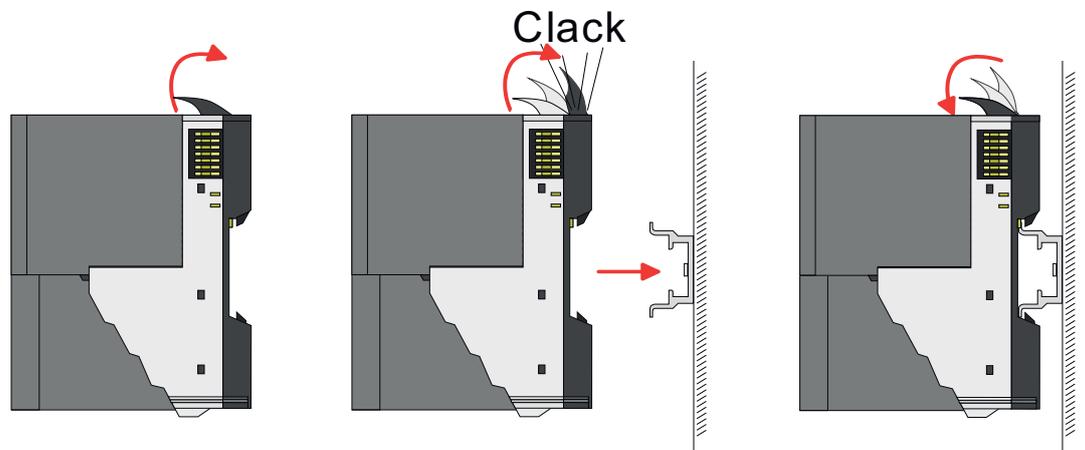


### CAUTION!

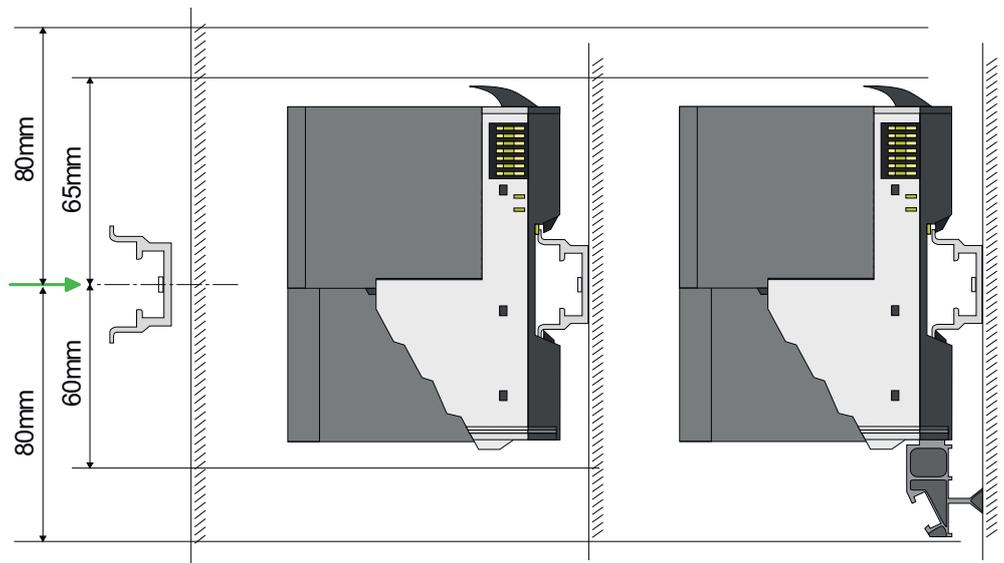
#### Requirements for UL compliance use

- Use for power supply exclusively SELV/PELV power supplies.
- The System SLIO must be installed and operated in a housing according to IEC 61010-1 9.3.2 c).

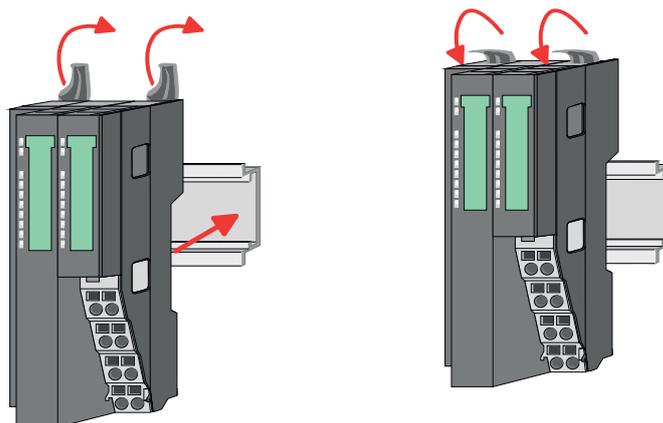
There are locking lever at the top side of the bus coupler. For mounting and demounting these locking lever are to be turned upwards until these engage. Place the bus coupler at the mounting rail. The bus coupler is fixed to the mounting rail by pushing downward the locking levers. The bus coupler is directly mounted at a mounting rail. Up to 64 modules may be mounted. The electronic and power section supply are connected via the back-plane bus. Please consider here that the sum current of the electronic power supply does not exceed the maximum value of 3A. By means of the power module 007-1AB10 the current of the electronic power supply may be expanded accordingly.



### Proceeding



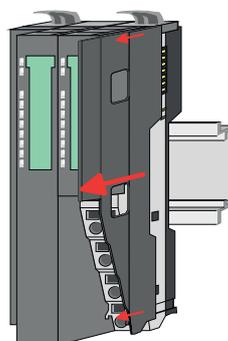
1. ➔ Mount the mounting rail! Please consider that a clearance from the middle of the mounting rail of at least 80mm above and 60mm below, respectively 80mm by deployment of shield bus carriers, exist.



2. Turn the locking lever upwards, place the bus coupler at the mounting rail and turn the lever downward.

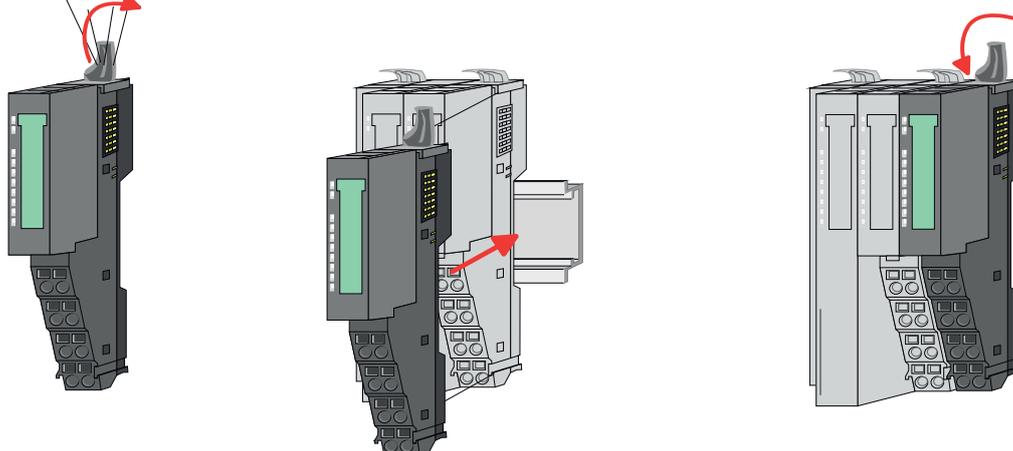
### Mounting peripheral modules

The procedure is identical for 8x and 16x peripheral modules.

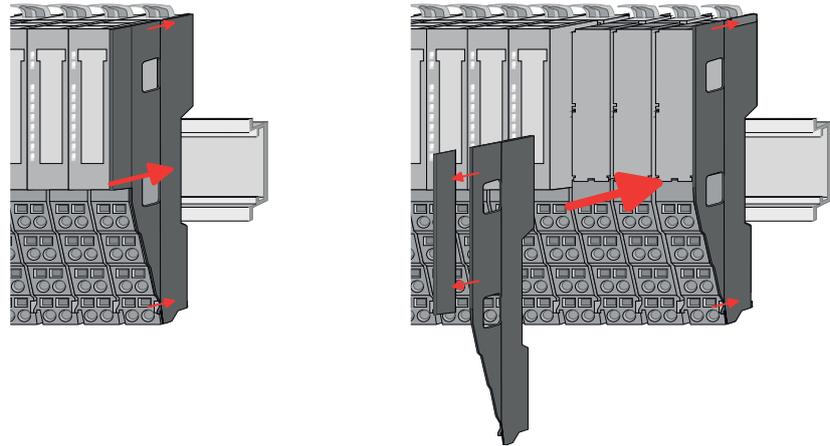


1. Before mounting the peripheral modules you have to remove the bus cover at the right side of the bus coupler by pulling it forward. Keep the cover for later mounting.

Clack



2. Mount the peripheral modules you want.



3. After mounting the whole system, to protect the backplane bus connectors at the last module you have to mount the bus cover, now. If the last module is a clamp module, for adaptation the upper part of the bus cover is to be removed.

## 2.5 Wiring



### CAUTION!

#### Consider temperature for external cables!

Cables may experience temperature increase due to system heat dissipation. Thus the cabling specification must be chosen 5°C above ambient temperature!



### CAUTION!

#### Separate insulation areas!

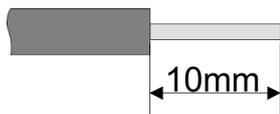
The system is specified for SELV/PELV environment. Devices, which are attached to the system must meet these specifications. Installation and cable routing other than SELV/PELV specification must be separated from the system's equipment!

### 2.5.1 Wiring bus coupler

#### Terminal module terminals

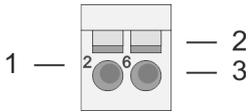
The System SLIO bus coupler have a power module integrated. Terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw terminal connections this type of connection is vibration proof.

#### Data

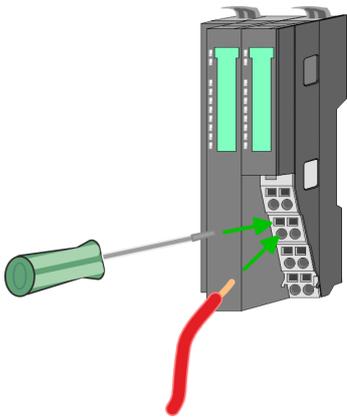
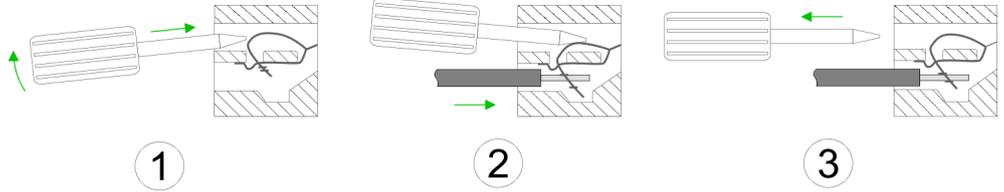


$U_{max}$	30V DC
$I_{max}$	10A
Cross section	0.08 ... 1.5mm <sup>2</sup> (AWG 28 ... 16)
Stripping length	10mm

**Wiring procedure**

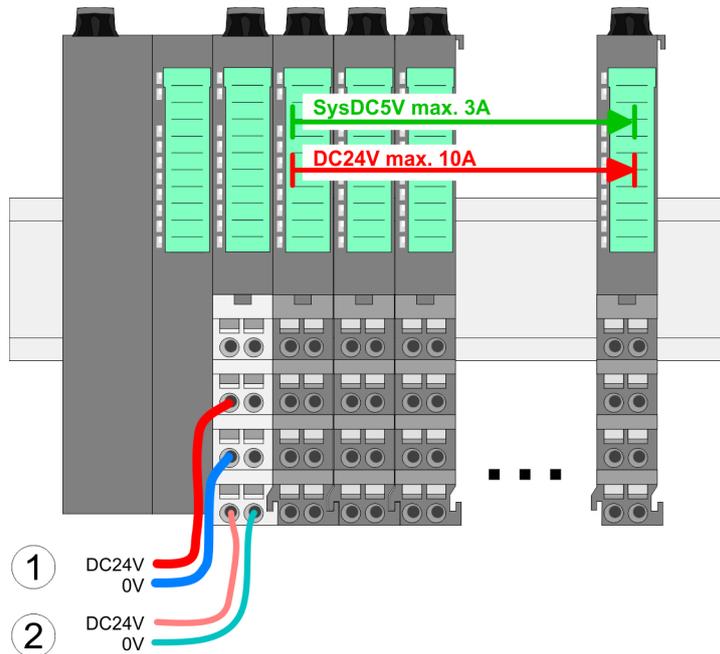


- 1 Pin number at the connector
- 2 Opening for screwdriver
- 3 Connection hole for wire



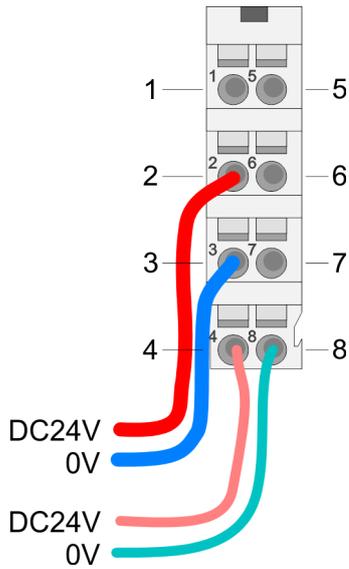
- 1. Insert a suited screwdriver at an angle into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
- 2. Insert the stripped end of wire into the round opening. You can use wires with a cross section of 0.08mm<sup>2</sup> up to 1.5mm<sup>2</sup>
- 3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

**Standard wiring**



- (1) DC 24V for power section supply I/O area (max. 10A)
- (2) DC 24V for electronic power supply bus coupler and I/O area

**PM - Power module**



For wires with a core cross-section of 0.08mm<sup>2</sup> up to 1.5mm<sup>2</sup>.

Pos.	Function	Type	Description
1	---	---	not connected
2	DC 24V	I	DC 24V for power section supply
3	0V	I	GND for power section supply
4	Sys DC 24V	I	DC 24V for electronic section supply
5	---	---	not connected
6	DC 24V	I	DC 24V for power section supply
7	0V	I	GND for power section supply
8	Sys 0V	I	GND for electronic section supply

I: Input



**CAUTION!**

Since the power section supply is not internally protected, it is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected by a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!



*The electronic power section supply is internally protected against higher voltage by fuse. The fuse is within the power module. If the fuse releases, its electronic module must be exchanged!*

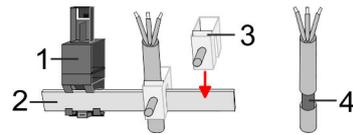
**Fusing**

- The power section supply is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected with a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!
- It is recommended to externally protect the electronic power supply for bus coupler and I/O area with a 2A fuse (fast) respectively by a line circuit breaker 2A characteristics Z.
- The electronic power supply for the I/O area of the power module 007-1AB10 should also be externally protected with a 1A fuse (fast) respectively by a line circuit breaker 1A characteristics Z.

**State of the electronic power supply via LEDs**

After PowerON of the System SLIO the LEDs RUN respectively MF get on so far as the sum current does not exceed 3A. With a sum current greater than 3A the LEDs may not be activated. Here the power module with the order number 007-1AB10 is to be placed between the peripheral modules.

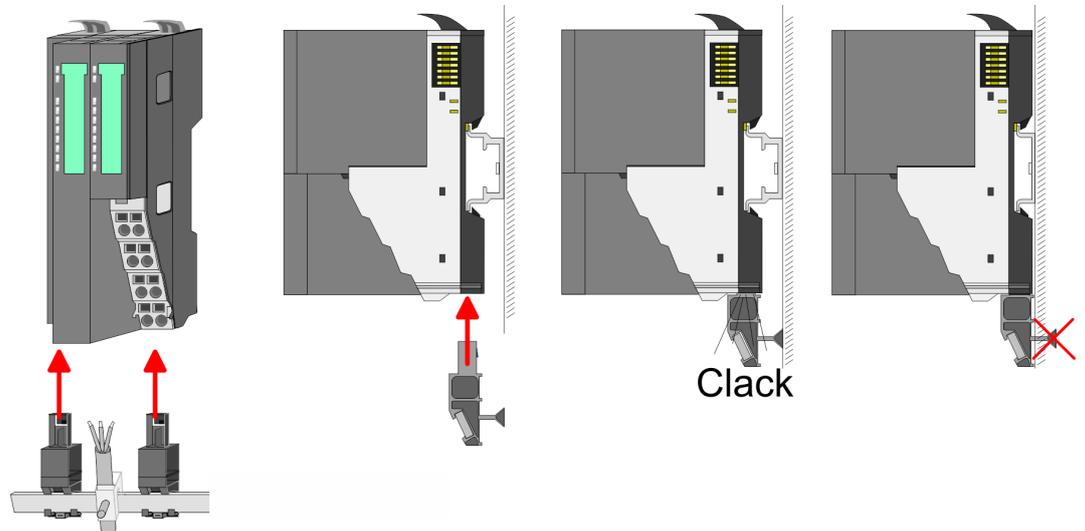
## Shield attachment



- 1 Shield bus carrier
- 2 Shield bus (10mm x 3mm)
- 3 Shield clamp
- 4 Cable shield

To attach the shield the mounting of shield bus carriers are necessary. The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

1. ➤ Each System SLIO module has a carrier hole for the shield bus carrier. Push the shield bus carrier, until they engage into the module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.
2. ➤ Put your shield bus into the shield bus carrier.



3. ➤ Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

## 2.5.2 Wiring 8x periphery modules

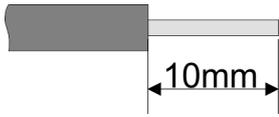
## Terminal module terminals

**CAUTION!****Do not connect hazardous voltages!**

If this is not explicitly stated in the corresponding module description, hazardous voltages are not allowed to be connected to the corresponding terminal module!

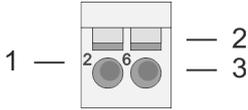
- With wiring the terminal modules, terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw terminal connections this type of connection is vibration proof.

Data

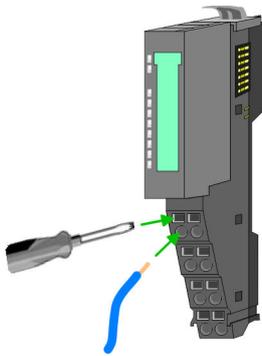
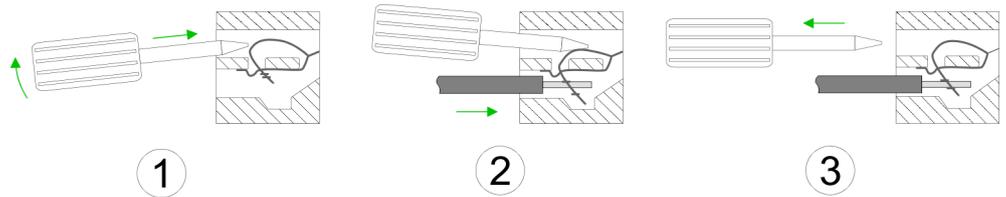


$U_{max}$  240V AC / 30V DC  
 $I_{max}$  10A  
 Cross section 0.08 ... 1.5mm<sup>2</sup> (AWG 28 ... 16)  
 Stripping length 10mm

Wiring procedure

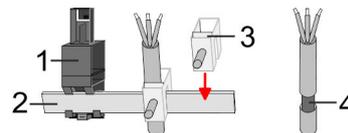


- 1 Pin number at the connector
- 2 Opening for screwdriver
- 3 Connection hole for wire



1. Insert a suited screwdriver at an angle into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
2. Insert the stripped end of wire into the round opening. You can use wires with a cross section of 0.08mm<sup>2</sup> up to 1.5mm<sup>2</sup>
3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

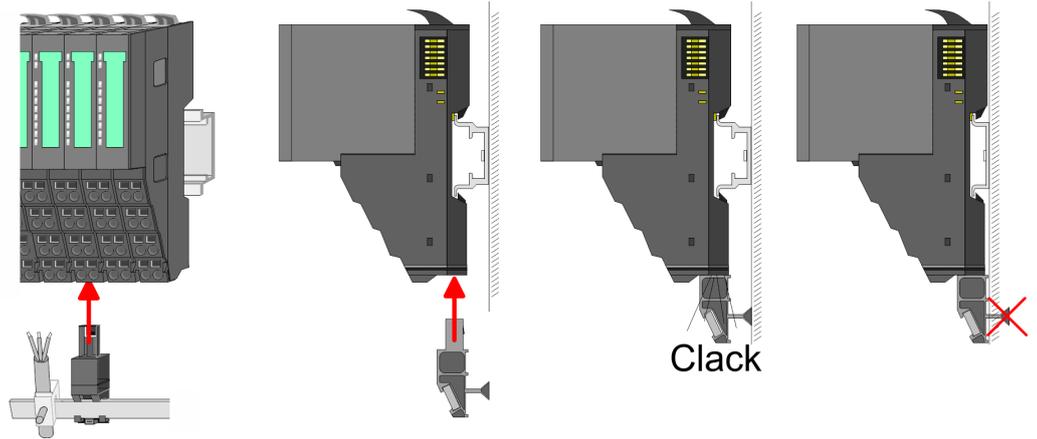
Shield attachment



- 1 Shield bus carrier
- 2 Shield bus (10mm x 3mm)
- 3 Shield clamp
- 4 Cable shield

To attach the shield the mounting of shield bus carriers are necessary. The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

1. Each System SLIO 8x periphery module has a carrier hole for the shield bus carrier. Push the shield bus carrier, until they engage into the module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.
2. Put your shield bus into the shield bus carrier.



3. Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

### 2.5.3 Wiring 16x periphery modules

#### Terminal block connectors



#### CAUTION!

#### Do not connect hazardous voltages!

If this is not explicitly stated in the corresponding module description, hazardous voltages are not allowed to be connected to the corresponding terminal block!

- The 16x periphery module has a removable terminal block for wiring.
- With the wiring of the terminal block a "push-in" spring-clip technique is used. This allows a quick and easy connection of your signal and supply lines.
- The clamping off takes place by means of a screwdriver.
- Please use copper wire only!

#### Data



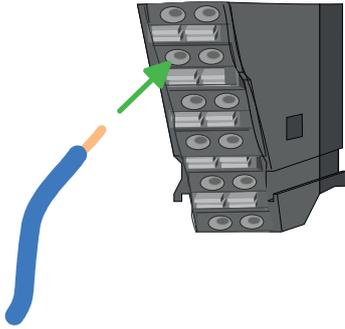
$U_{max}$	30V DC
$I_{max}$	10A
Cross section solid wire	0.25 ... 0.75mm <sup>2</sup>
Cross section with ferrule	0.14 ... 0.75mm <sup>2</sup>
Wire type	CU
AWG	24 ... 16
Stripping length	10mm

#### Wiring procedure



- 1 Release area
- 2 Connection hole for wire

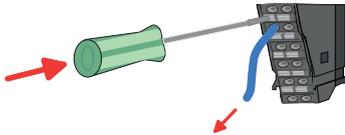
**Insert wire**



The wiring happens without a tool.

1. Determine according to the casing labelling the connection position.
2. Insert through the round connection hole of the according contact your prepared wire until it stops, so that it is fixed.
  - ⇒ By pushing the contact spring opens, thus ensuring the necessary contact pressure.

**Remove wire**



The wire is to be removed by means of a screwdriver with 2.5mm blade width.

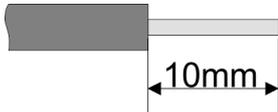
1. Press with your screwdriver vertically at the release button.
  - ⇒ The contact spring releases the wire.
2. Pull the wire from the round hole.

**2.5.4 Wiring power modules**

**Terminal module terminals**

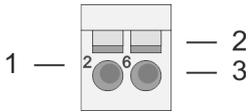
Power modules are either integrated to the head module or may be installed between the periphery modules. With power modules, terminals with spring clamp technology are used for wiring. The spring clamp technology allows quick and easy connection of your signal and supply lines. In contrast to screw terminal connections this type of connection is vibration proof.

**Data**

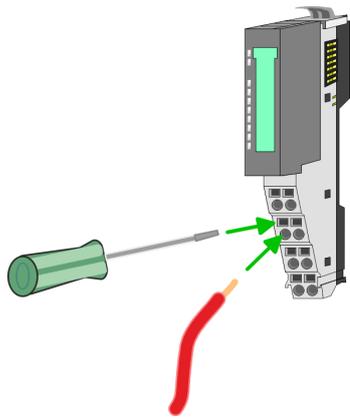
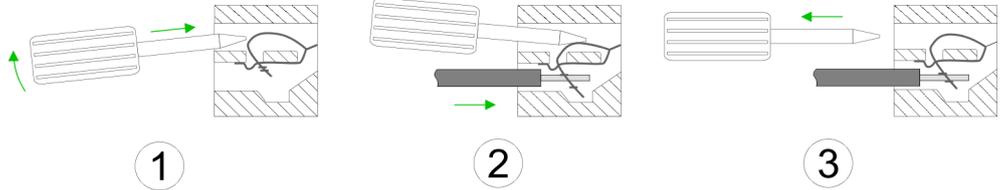


$U_{max}$	30V DC
$I_{max}$	10A
Cross section	0.08 ... 1.5mm <sup>2</sup> (AWG 28 ... 16)
Stripping length	10mm

**Wiring procedure**

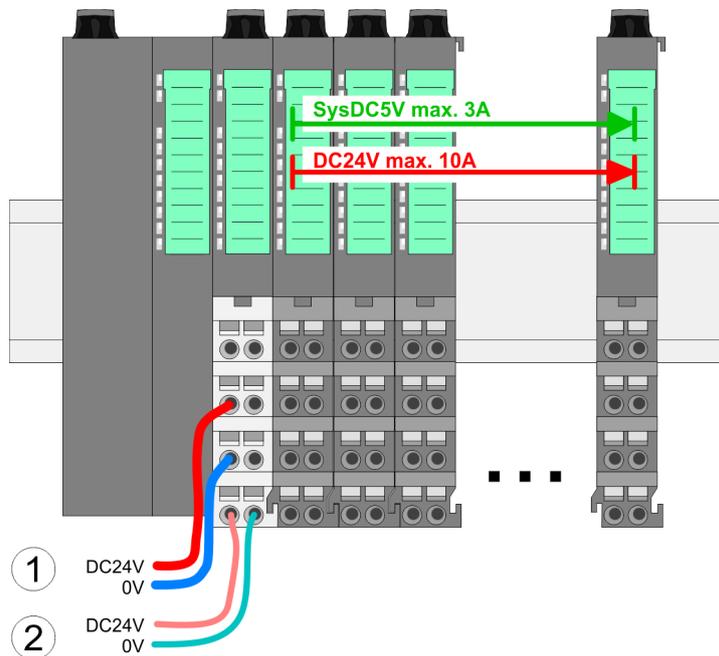


- 1 Pin number at the connector
- 2 Opening for screwdriver
- 3 Connection hole for wire



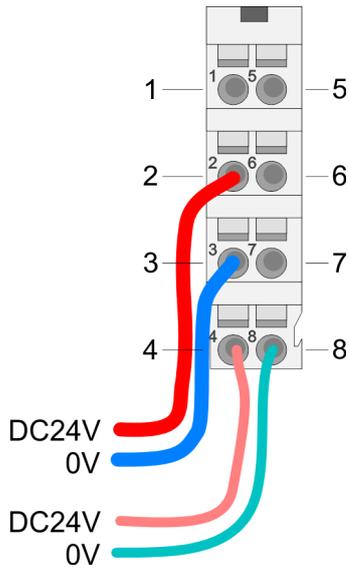
- 1. Insert a suited screwdriver at an angle into the square opening as shown. Press and hold the screwdriver in the opposite direction to open the contact spring.
- 2. Insert the stripped end of wire into the round opening. You can use wires with a cross section of 0.08mm<sup>2</sup> up to 1.5mm<sup>2</sup>
- 3. By removing the screwdriver, the wire is securely fixed via the spring contact to the terminal.

**Standard wiring**



- (1) DC 24V for power section supply I/O area (max. 10A)
- (2) DC 24V for electronic power supply bus coupler and I/O area

**PM - Power module**



For wires with a core cross-section of 0.08mm<sup>2</sup> up to 1.5mm<sup>2</sup>.

Pos.	Function	Type	Description
1	---	---	not connected
2	DC 24V	I	DC 24V for power section supply
3	0V	I	GND for power section supply
4	Sys DC 24V	I	DC 24V for electronic section supply
5	---	---	not connected
6	DC 24V	I	DC 24V for power section supply
7	0V	I	GND for power section supply
8	Sys 0V	I	GND for electronic section supply

I: Input



**CAUTION!**

Since the power section supply is not internally protected, it is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected by a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!



*The electronic power section supply is internally protected against higher voltage by fuse. The fuse is within the power module. If the fuse releases, its electronic module must be exchanged!*

**Fusing**

- The power section supply is to be externally protected with a fuse, which corresponds to the maximum current. This means max. 10A is to be protected with a 10A fuse (fast) respectively by a line circuit breaker 10A characteristics Z!
- It is recommended to externally protect the electronic power supply for head modules and I/O area with a 2A fuse (fast) respectively by a line circuit breaker 2A characteristics Z.
- The electronic power supply for the I/O area of the power module 007-1AB10 should also be externally protected with a 1A fuse (fast) respectively by a line circuit breaker 1A characteristics Z.

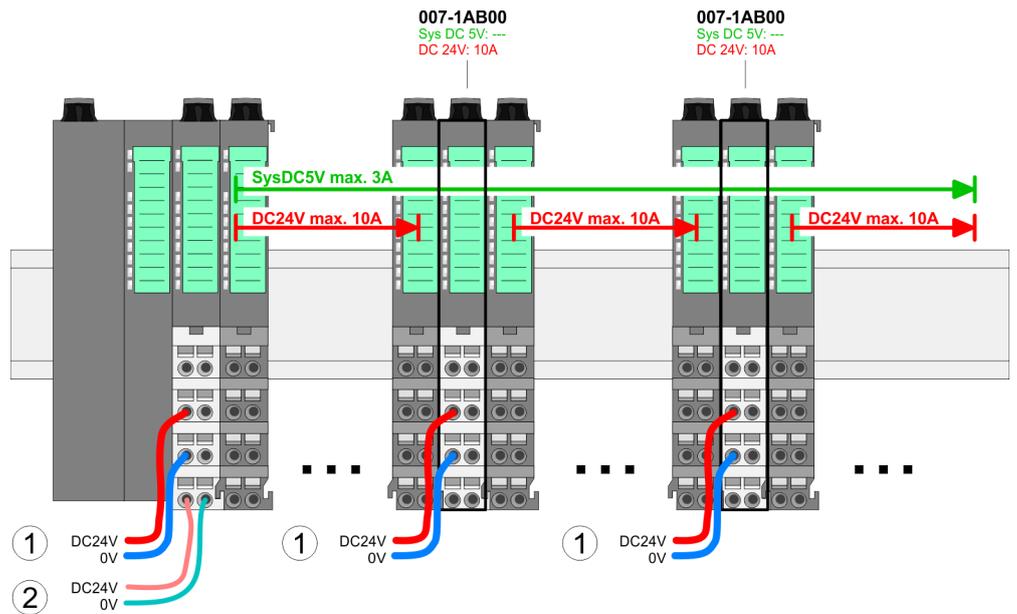
**State of the electronic power supply via LEDs**

After PowerON of the System SLIO the LEDs RUN respectively MF get on so far as the sum current does not exceed 3A. With a sum current greater than 3A the LEDs may not be activated. Here the power module with the order number 007-1AB10 is to be placed between the peripheral modules.

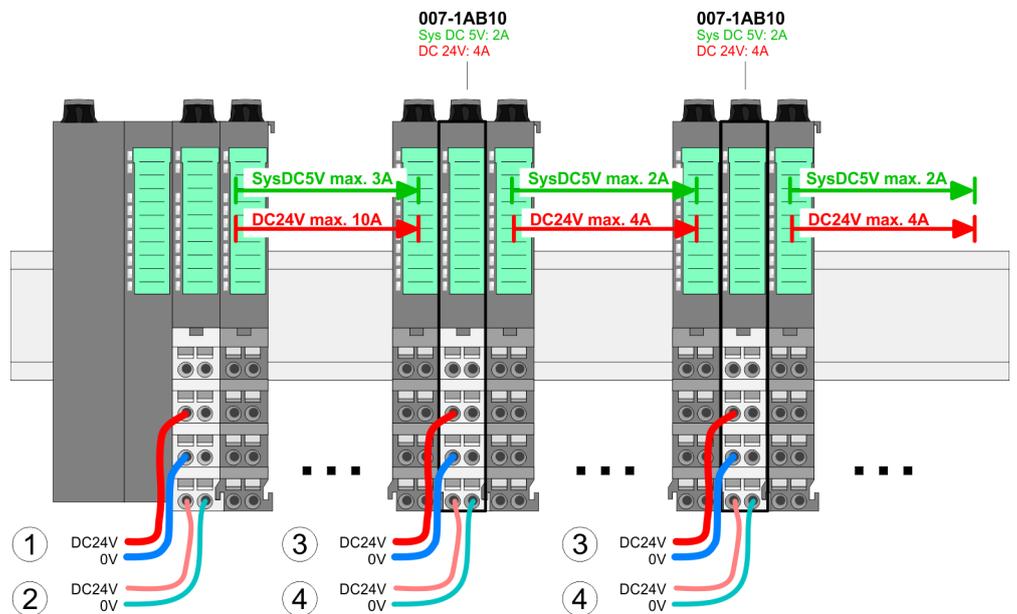
**Deployment of the power modules**

- If the 10A for the power section supply is no longer sufficient, you may use the power module with the order number 007-1AB00. So you have also the possibility to define isolated groups.
- The power module with the order number 007-1AB10 is to be used if the 3A for the electronic power supply at the backplane bus is no longer sufficient. Additionally you get an isolated group for the DC 24V power section supply with max. 4A.
- By placing the power module 007-1AB10 at the following backplane bus modules may be placed with a sum current of max. 2A. Afterwards a power module is to be placed again. To secure the power supply, the power modules may be mixed used.

**Power module 007-1AB00**

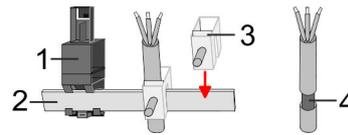


**Power module 007-1AB10**



- (1) DC 24V for power section supply I/O area (max. 10A)
- (2) DC 24V for electronic power supply bus coupler and I/O area
- (3) DC 24V for power section supply I/O area (max. 4A)
- (4) DC 24V for electronic power supply I/O area

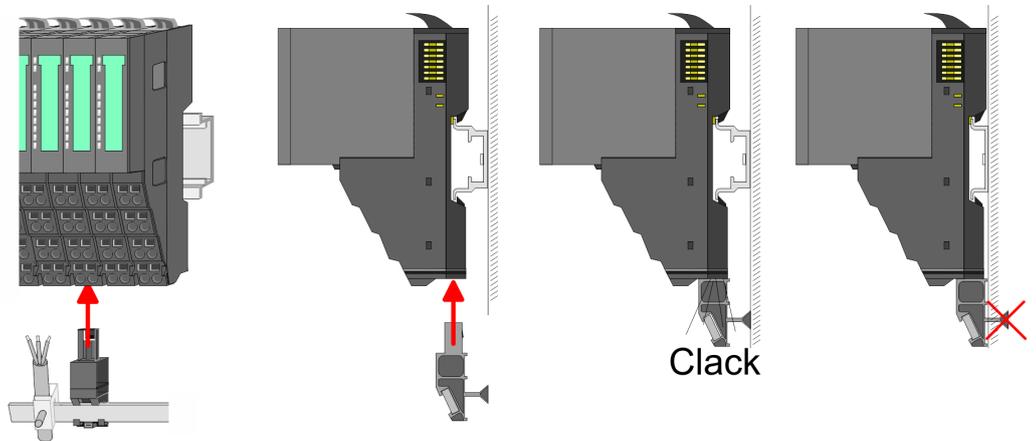
Shield attachment



- 1 Shield bus carrier
- 2 Shield bus (10mm x 3mm)
- 3 Shield clamp
- 4 Cable shield

To attach the shield the mounting of shield bus carriers are necessary. The shield bus carrier (available as accessory) serves to carry the shield bus to connect cable shields.

- 1. ➤ Each System SLIO 8x peripheral module has a carrier hole for the shield bus carrier. Push the shield bus carrier, until they engage into the module. With a flat mounting rail for adaptation to a flat mounting rail you may remove the spacer of the shield bus carrier.
- 2. ➤ Put your shield bus into the shield bus carrier.



- 3. ➤ Attach the cables with the accordingly stripped cable screen and fix it by the shield clamp with the shield bus.

2.6 Demounting

2.6.1 Demounting bus coupler

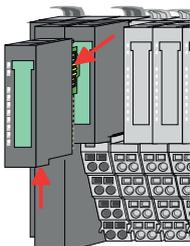
Proceeding



**CAUTION!**

Bus interface and power module may not be separated! Here you may only exchange the electronic module!

- 1. ➤ Power-off your system.
- 2. ➤ Remove if exists the wiring of the bus coupler.

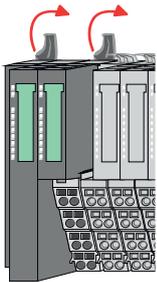


- 3. ➤

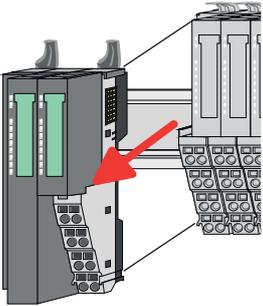


*For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*

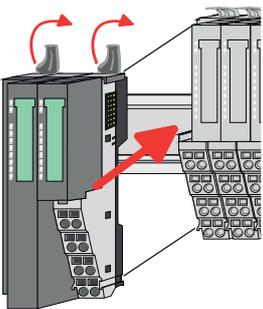
Press the unlocking lever at the lower side of the just mounted right module near the bus coupler and pull it forward.



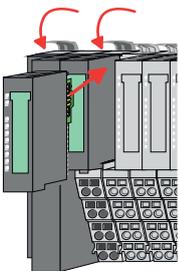
- 4.** Turn all the locking lever of the bus coupler to be exchanged upwards.



- 5.** Pull the bus coupler forward.  
**6.** For mounting turn all the locking lever of the bus coupler to be exchanged upwards.



- 7.** To mount the bus coupler put it to the left periphery module and push it, guided by the stripes, to the mounting rail.  
**8.** Turn all the locking lever downward, again.



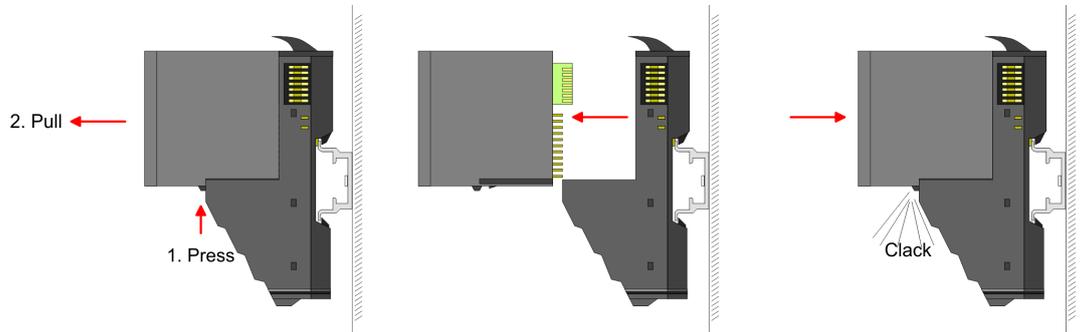
- 9.** Plug again the electronic module, which you have removed before.  
**10.** Wire your bus coupler.  
⇒ Now you can bring your system back into operation.

## 2.6.2 Demounting 8x periphery modules

### Proceeding

#### Exchange of an electronic module

1. ➤ Power-off your system.



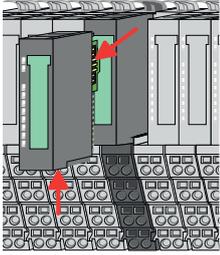
2. ➤ For the exchange of a electronic module, the electronic module may be pulled forward after pressing the unlocking lever at the lower side of the module.
3. ➤ For installation plug the new electronic module guided by the strips at the lower side until this engages to the terminal module.
  - ⇒ Now you can bring your system back into operation.



#### **Easy Maintenance**

*'Easy Maintenance' means the support for adding and removing electronic modules during operation without having to restart the system. If this is supported by your head module, you will find more detailed information on this in the "Deployment" chapter.*

### Exchange of a periphery module

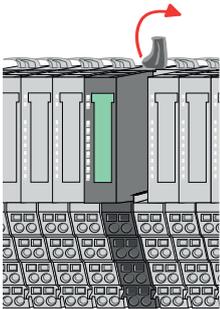


1. ➤ Power-off your system.
2. ➤ Remove if exists the wiring of the module.
3. ➤

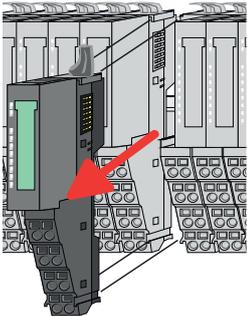


*For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module right beside. After mounting it may be plugged again.*

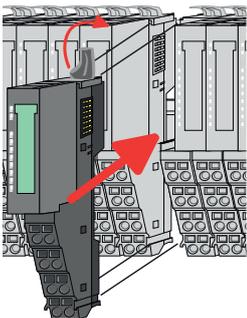
Press the unlocking lever at the lower side of the just mounted right module and pull it forward.



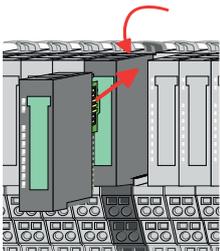
4. ➤ Turn the locking lever of the module to be exchanged upwards.



5. ➤ Pull the module.
6. ➤ For mounting turn the locking lever of the module to be mounted upwards.

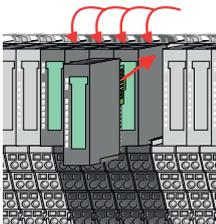
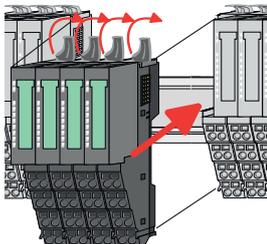
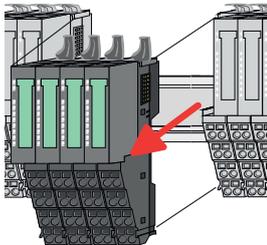
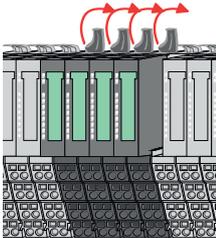
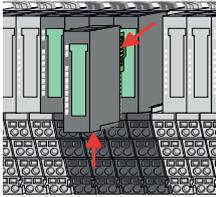


7. ➤ To mount the module put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.
8. ➤ Turn the locking lever downward, again.



9. ➤ Plug again the electronic module, which you have removed before.
10. ➤ Wire your module.
  - ⇒ Now you can bring your system back into operation.

**Exchange of a module group**



1. ➤ Power-off your system.
2. ➤ Remove if exists the wiring of the module group.

3. ➤

**i** For demounting and exchange of a (head) module or a group of modules, due to mounting reasons you always have to remove the electronic module *right beside*. After mounting it may be plugged again.

Press the unlocking lever at the lower side of the just mounted right module near the module group and pull it forward.

4. ➤ Turn all the locking lever of the module group to be exchanged upwards.

5. ➤ Pull the module group forward.

6. ➤ For mounting turn all the locking lever of the module group to be mounted upwards.

7. ➤ To mount the module group put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.

8. ➤ Turn all the locking lever downward, again.

9. ➤ Plug again the electronic module, which you have removed before.

10. ➤ Wire your module group.

⇒ Now you can bring your system back into operation.

### 2.6.3 Demounting 16x periphery modules

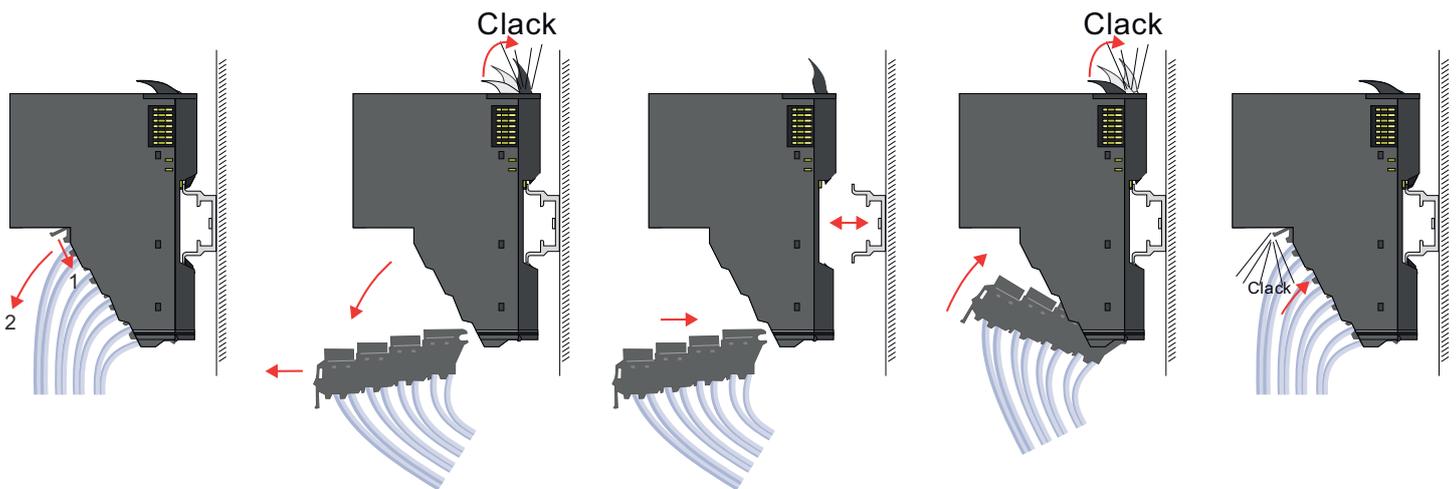
#### Proceeding

#### Exchange of an electronic unit

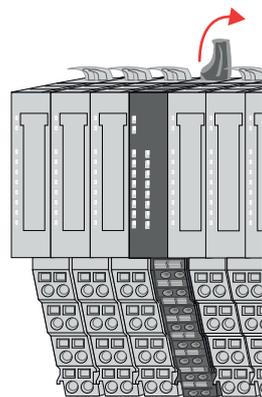
1. ➤ Power-off your system.
2. ➤ To replace an electronic unit, you can push down and pull off the terminal block after releasing the lock.

To mount the terminal block, place it horizontally on the lower side of the electronic unit and push it towards the electronic unit until it clicks into place.

⇒ Now you can bring your system back into operation.



#### Exchange of a 16x periphery module

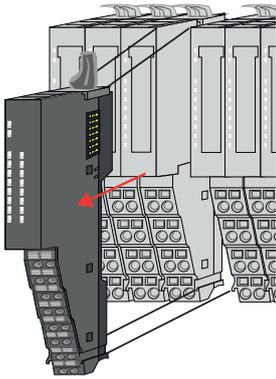


1. ➤ Power-off your system.
2. ➤ Remove if exists the wiring of the module respectively the wired terminal block.
3. ➤

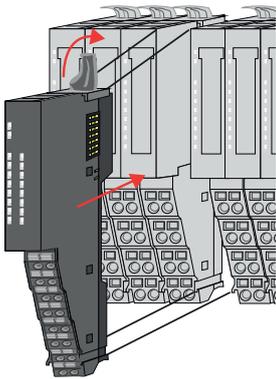


*In contrast to 8x periphery modules, you can directly demount and mount 16x periphery modules.*

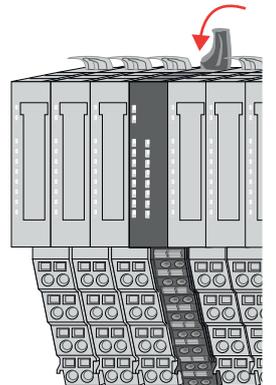
Turn the locking lever of the module to be exchanged upwards.



4. ➤ Pull the module.
5. ➤ For mounting turn the locking lever of the module to be mounted upwards.

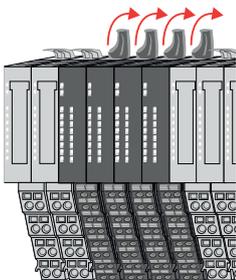


6. ➤ To mount the module put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.



7. ➤ Turn the locking lever downward, again.
8. ➤ Wire your module respectively plug the wired terminal block again.
  - ⇒ Now you can bring your system back into operation.

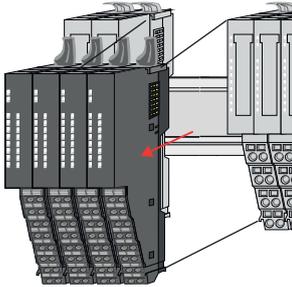
**Exchange of a module group**



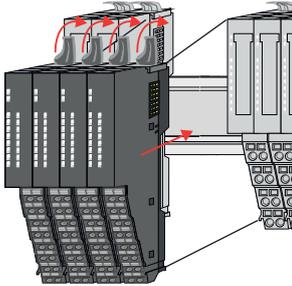
1. ➤ Power-off your system.
2. ➤ Remove if exists the wiring of the module group respectively the wired terminal blocks.
3. ➤

**i** *In contrast to 8x periphery modules, you can directly demount and mount 16x periphery modules.*

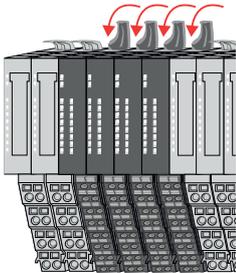
Turn all the locking lever of the module group to be exchanged upwards.



4. ➤ Pull the module group forward.
5. ➤ For mounting turn all the locking lever of the module group to be mounted upwards.



6. ➤ To mount the module group put it to the gap between the both modules and push it, guided by the stripes at both sides, to the mounting rail.



7. ➤ Turn all the locking lever downward, again.
8. ➤ Wire your module group respectively plug the wired terminal blocks again.  
⇒ Now you can bring your system back into operation.

## 2.7 Trouble shooting - LEDs

### General

Each module has the LEDs RUN and MF on its front side. Errors or incorrect modules may be located by means of these LEDs.

In the following illustrations flashing LEDs are marked by ☼.

### Sum current of the electronic power supply exceeded

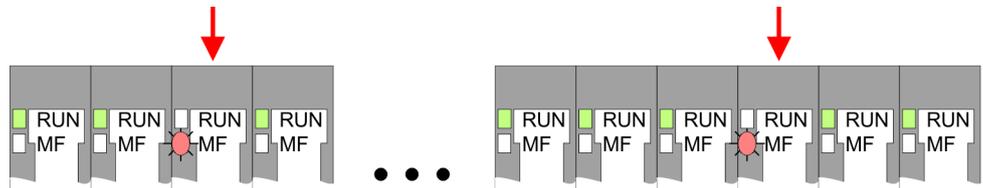


**Behaviour:** After PowerON the RUN LED of each module is off and the MF LED of each module is sporadically on.

**Reason:** The maximum current for the electronic power supply is exceeded.

**Remedy:** As soon as the sum current of the electronic power supply is exceeded, always place the power module 007-1AB10. ↪ Chap. 2.5.4 'Wiring power modules' page 26

### Error in configuration

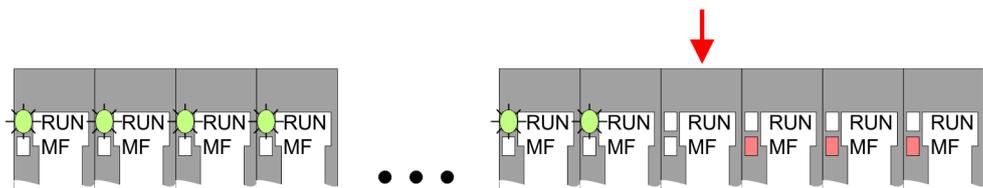


**Behaviour:** After PowerON the MF LED of one module respectively more modules flashes. The RUN LED remains off.

**Reason:** At this position a module is placed, which does not correspond to the configured module.

**Remedy:** Match configuration and hardware structure.

### Module failure



**Behaviour:** After PowerON all of the RUN LEDs up to the defective module are flashing. With all following modules the MF LED is on and the RUN LED is off.

**Reason:** The module on the right of the flashing modules is defective.

**Remedy:** Replace the defective module.

## 2.8 Industrial security and installation guidelines

### 2.8.1 Industrial security in information technology

#### Latest version

This chapter can also be found as a guide '*Industrial IT Security*' at [www.yaskawa.eu.com](http://www.yaskawa.eu.com)

#### Hazards

The topic of data security and access protection has become increasingly important in the industrial environment. The increased networking of entire industrial systems to the network levels within the company together with the functions of remote maintenance have all served to increase vulnerability. Hazards can arise from:

- Internal manipulation such as technical errors, operating and program errors and deliberate program or data manipulation.
- External manipulation such as software viruses, worms and Trojans.
- Human carelessness such as password phishing.

#### Precautions

The most important precautions to prevent manipulation and loss of data security in the industrial environment are:

- Encrypting the data traffic by means of certificates.
- Filtering and inspection of the traffic by means of VPN - "Virtual Private Networks".
- Identification of the user by "Authentication" via safe channels.
- Segmenting in protected automation cells, so that only devices in the same group can exchange data.
- Deactivation of unnecessary hardware and software.

#### Further Information

You can find more information about the measures on the following websites:

- Federal Office for Information Technology [www.bsi.bund.de](http://www.bsi.bund.de)
- Cybersecurity & Infrastructure Security Agency [us-cert.cisa.gov](http://us-cert.cisa.gov)
- VDI / VDE Society for Measurement and Automation Technology [www.vdi.de](http://www.vdi.de)

### 2.8.1.1 Protection of hardware and applications

#### Precautions

- Do not integrate any components or systems into public networks.
  - Use VPN "Virtual Private Networks" for use in public networks. This allows you to control and filter the data traffic accordingly.
- Always keep your system up-to-date.
  - Always use the latest firmware version for all devices.
  - Update your user software regularly.
- Protect your systems with a firewall.
  - The firewall protects your infrastructure internally and externally.
  - This allows you to segment your network and isolate entire areas.
- Secure access to your plants via user accounts.
  - If possible, use a central user management system.
  - Create a user account for each user for whom authorization is essential.
  - Always keep user accounts up-to-date and deactivate unused user accounts.
- Secure access to your plants via secure passwords.
  - Change the password of a standard login after the first start.
  - Use strong passwords consisting of upper/lower case, numbers and special characters. The use of a password generator or manager is recommended.
  - Change the passwords according to the rules and guidelines that apply to your application.
- Deactivate inactive communication ports respectively protocols.
  - Only the communication ports that are used for communication should be activated.
  - Only the communication protocols that are used for communication should be activated.
- Consider possible defence strategies when planning and securing the system.
  - The isolation of components alone is not sufficient for comprehensive protection. An overall concept is to be drawn up here, which also provides defensive measures in the event of a cyber attack.
  - Periodically carry out threat assessments. Among others, a comparison is made here between the protective measures taken and those required.
- Limit the use of external storage media.
  - Via external storage media such as USB memory sticks or SD memory cards, malware can get directly into a system while bypassing a firewall.
  - External storage media or their slots must be protected against unauthorized physical access, e.g. by using a lockable control cabinet.
  - Make sure that only authorized persons have access.
  - When disposing of storage media, make sure that they are safely destroyed.
- Use secure access paths such as HTTPS or VPN for remote access to your plant.
- Enable security-related event logging in accordance with the applicable security policy and legal requirements for data protection.

### 2.8.1.2 Protection of PC-based software

#### Precautions

Since PC-based software is used for programming, configuration and monitoring, it can also be used to manipulate entire systems or individual components. Particular caution is required here!

- Use user accounts on your PC systems.
  - If possible, use a central user management system.
  - Create a user account for each user for whom authorization is essential.
  - Always keep user accounts up-to-date and deactivate unused user accounts.
- Protect your PC systems with secure passwords.
  - Change the password of a standard login after the first start.
  - Use strong passwords consisting of upper/lower case, numbers and special characters. The use of a password generator or manager is recommended.
  - Change the passwords according to the rules and guidelines that apply to your application.
- Enable security-related event logging in accordance with the applicable security policy and legal requirements for data protection.
- Protect your PC systems by security software.
  - Install virus scanners on your PC systems to identify viruses, trojans and other malware.
  - Install software that can detect phishing attacks and actively prevent them.
- Always keep your software up-to-date.
  - Update your operating system regularly.
  - Update your software regularly.
- Make regular backups and store the media at a safe place.
- Regularly restart your PC systems. Only boot from storage media that are protected against manipulation.
- Use encryption systems on your storage media.
- Perform security assessments regularly to reduce the risk of manipulation.
- Use only data and software from approved sources.
- Uninstall software which is not used.
- Disable unused services.
- Activate a password-protected screen lock on your PC systems.
- Always lock your PC systems as soon as you leave your PC workstation.
- Do not click any links that come from unknown sources. If necessary ask, e.g. on e-mails.
- Use secure access paths such as HTTPS or VPN for remote access to your PC system.

### 2.8.2 Installation guidelines

#### General

The installation guidelines contain information about the interference free deployment of a PLC system. There is the description of the ways, interference may occur in your PLC, how you can make sure the electromagnetic compatibility (EMC), and how you manage the isolation.

#### What does EMC mean?

Electromagnetic compatibility (EMC) means the ability of an electrical device, to function error free in an electromagnetic environment without being interfered respectively without interfering the environment.

The components are developed for the deployment in industrial environments and meets high demands on the EMC. Nevertheless you should project an EMC planning before installing the components and take conceivable interference causes into account.

**Possible interference causes**

Electromagnetic interferences may interfere your control via different ways:

- Electromagnetic fields (RF coupling)
- Magnetic fields with power frequency
- Bus system
- Power supply
- Protected earth conductor

Depending on the spreading medium (lead bound or lead free) and the distance to the interference cause, interferences to your control occur by means of different coupling mechanisms.

There are:

- galvanic coupling
- capacitive coupling
- inductive coupling
- radiant coupling

**Basic rules for EMC**

In the most times it is enough to take care of some elementary rules to guarantee the EMC. Please regard the following basic rules when installing your PLC.

- Take care of a correct area-wide grounding of the inactive metal parts when installing your components.
  - Install a central connection between the ground and the protected earth conductor system.
  - Connect all inactive metal extensive and impedance-low.
  - Please try not to use aluminium parts. Aluminium is easily oxidizing and is therefore less suitable for grounding.
- When cabling, take care of the correct line routing.
  - Organize your cabling in line groups (high voltage, current supply, signal and data lines).
  - Always lay your high voltage lines and signal respectively data lines in separate channels or bundles.
  - Route the signal and data lines as near as possible beside ground areas (e.g. suspension bars, metal rails, tin cabinet).
- Proof the correct fixing of the lead isolation.
  - Data lines must be shielded.
  - Analog lines must be shielded. When transmitting signals with small amplitudes the one sided laying of the isolation may be favourable.
  - Cables for frequency inverters, servo and stepper motors must be shielded.
  - Lay the line isolation extensively on an isolation/protected earth conductor rail directly after the cabinet entry and fix the isolation with cable clamps.
  - Make sure that the isolation/protected earth conductor rail is connected impedance-low with the cabinet.
  - Use metallic or metallised plug cases for isolated data lines.
- In special use cases you should appoint special EMC actions.
  - Consider to wire all inductivities with erase links.
  - Please consider luminescent lamps can influence signal lines.
- Create a homogeneous reference potential and ground all electrical operating supplies when possible.
  - Please take care for the targeted employment of the grounding actions. The grounding of the PLC serves for protection and functionality activity.
  - Connect installation parts and cabinets with your PLC in star topology with the isolation/protected earth conductor system. So you avoid ground loops.
  - If there are potential differences between installation parts and cabinets, lay sufficiently dimensioned potential compensation lines.

**Isolation of conductors**

Electrical, magnetically and electromagnetic interference fields are weakened by means of an isolation, one talks of absorption. Via the isolation rail, that is connected conductive with the rack, interference currents are shunt via cable isolation to the ground. Here you have to make sure, that the connection to the protected earth conductor is impedance-low, because otherwise the interference currents may appear as interference cause.

When isolating cables you have to regard the following:

- If possible, use only cables with isolation tangle.
- The hiding power of the isolation should be higher than 80%.
- Normally you should always lay the isolation of cables on both sides. Only by means of the both-sided connection of the isolation you achieve high quality interference suppression in the higher frequency area. Only as exception you may also lay the isolation one-sided. Then you only achieve the absorption of the lower frequencies. A one-sided isolation connection may be convenient, if:
  - the conduction of a potential compensating line is not possible.
  - analog signals (some mV respectively  $\mu\text{A}$ ) are transferred.
  - foil isolations (static isolations) are used.
- With data lines always use metallic or metallised plugs for serial couplings. Fix the isolation of the data line at the plug rack. Do not lay the isolation on the PIN 1 of the plug bar!
- At stationary operation it is convenient to strip the insulated cable interruption free and lay it on the isolation/protected earth conductor line.
- To fix the isolation tangles use cable clamps out of metal. The clamps must clasp the isolation extensively and have well contact.
- Lay the isolation on an isolation rail directly after the entry of the cable in the cabinet. Lead the isolation further on to your PLC and don't lay it on there again!

**CAUTION!****Please regard at installation!**

At potential differences between the grounding points, there may be a compensation current via the isolation connected at both sides.

Remedy: Potential compensation line

**2.9 General data for the System SLIO****Conformity and approval**

Conformity		
CE	2014/35/EU	Low-voltage directive
	2014/30/EU	EMC directive
Approval		
UL	-	Refer to Technical data
Others		
RoHS	2011/65/EU	Restriction of the use of certain hazardous substances in electrical and electronic equipment

General data for the System SLIO

**Protection of persons and device protection**

Type of protection	-	IP20
Electrical isolation		
to the field bus	-	electrically isolated
to the process level	-	electrically isolated
Insulation resistance	-	-
Insulation voltage to reference earth		
Inputs / outputs	-	AC / DC 50V, test voltage AC 500V
Protective measures	-	against short circuit

**Environmental conditions to EN 61131-2**

Climatic		
Storage / transport	EN 60068-2-14	-25...+70°C
Operation		
Horizontal installation hanging	EN 61131-2	0...+60°C
Horizontal installation lying	EN 61131-2	0...+55°C
Vertical installation	EN 61131-2	0...+50°C
Air humidity	EN 60068-2-30	RH1 (without condensation, rel. humidity 10...95%)
Pollution	EN 61131-2	Degree of pollution 2
Installation altitude max.	-	2000m
Mechanical		
Oscillation	EN 60068-2-6	1g, 9Hz ... 150Hz
Shock	EN 60068-2-27	15g, 11ms

**Mounting conditions**

Mounting place	-	In the control cabinet
Mounting position	-	Horizontal and vertical

EMC	Standard	Comment	
Emitted interference	EN 61000-6-4	Class A (Industrial area)	
Noise immunity zone B	EN 61000-6-2	Industrial area	
		EN 61000-4-2	ESD 8kV at air discharge (degree of severity 3), 4kV at contact discharge (degree of severity 2)
		EN 61000-4-3	HF field immunity (casing) 80MHz ... 1000MHz, 10V/m, 80% AM (1kHz) 1.4GHz ... 2.0GHz, 3V/m, 80% AM (1kHz) 2GHz ... 2.7GHz, 1V/m, 80% AM (1kHz)
		EN 61000-4-6	HF conducted 150kHz ... 80MHz, 10V, 80% AM (1kHz)
		EN 61000-4-4	Burst, degree of severity 3
	EN 61000-4-5	Surge, degree of severity 3 <sup>1</sup>	

1) Due to the high-energetic single pulses with Surge an appropriate external protective circuit with lightning protection elements like conductors for lightning and overvoltage is necessary.

## 2.9.1 Use in difficult operating conditions



*Without additional protective measures, the products must not be used in locations with difficult operating conditions; e.g. due to:*

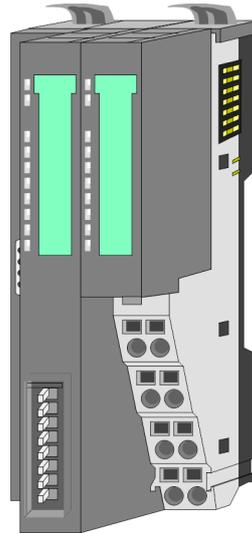
- *dust generation*
- *chemically active substances (corrosive vapors or gases)*
- *strong electric or magnetic fields*

## 3 Hardware description

### 3.1 Properties

#### Features

- 16 Rx and 16 Tx PDOs
- 2 SDOs
- Support of every transfer rates
- PDO linking
- PDO mapping: variable
- CAN bus address setting via DIP switch

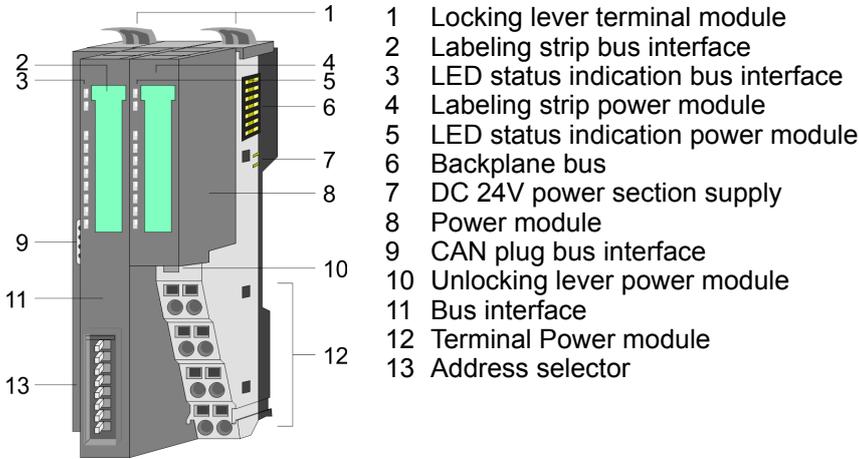


#### Order data

Type	Order number	Description
IM 053CAN	053-1CA00	CAN slave for System SLIO

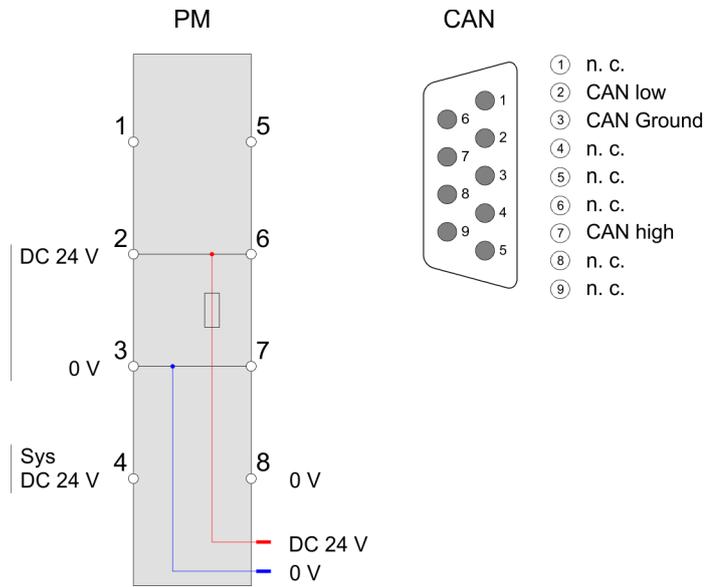
### 3.2 Structure

#### 053-1CA00



### 3.2.1 Interfaces

#### Interfaces

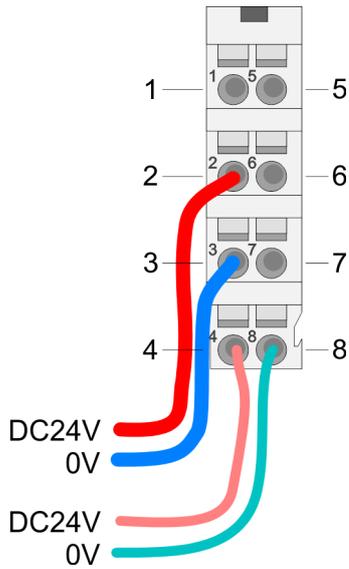


**CAUTION!**

Bus interface and power module of the bus coupler may not be separated!

Here you may only exchange the electronic module!

**PM - Power module**



For wires with a core cross-section of 0.08mm<sup>2</sup> up to 1.5mm<sup>2</sup>.

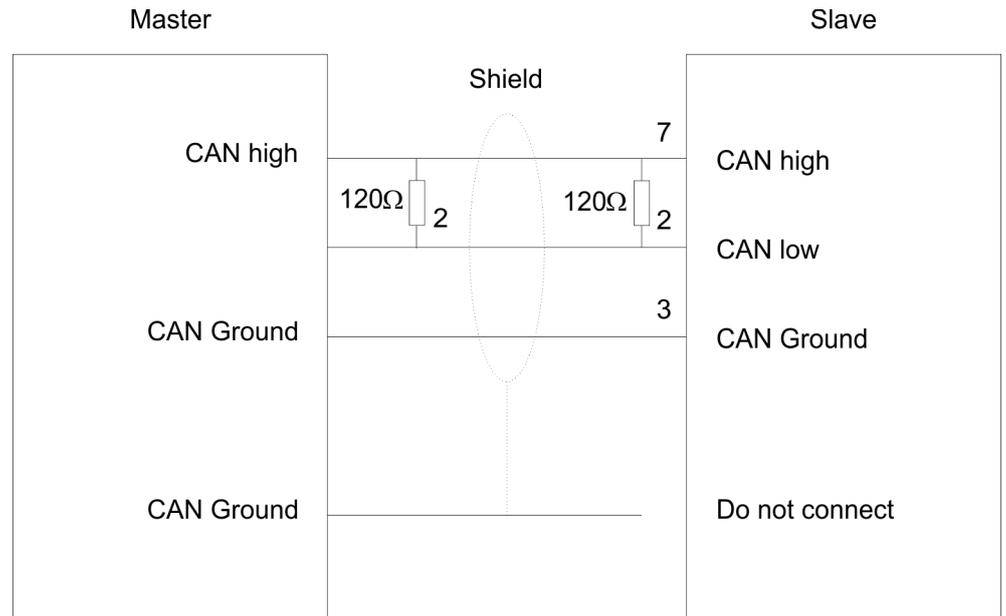
Pos.	Function	Type	Description
1	---	---	not connected
2	DC 24V	I	DC 24V for power section supply
3	0V	I	GND for power section supply
4	Sys DC 24V	I	DC 24V for electronic section supply
5	---	---	not connected
6	DC 24V	I	DC 24V for power section supply
7	0V	I	GND for power section supply
8	Sys 0V	I	GND for electronic section supply

I: Input

**Bus connection**

*9 pin CAN plug:*

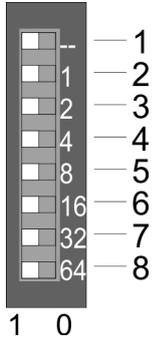
The CAN bus communication medium is a screened three-core cable. All stations on systems having more than two stations are wired in parallel. This means that the bus cable must be looped from station to station without interruptions.



*The end of the bus cable must be terminated with a 120Ω terminating resistor to prevent reflections and the associated communication errors!*

### 3.2.2 Address selector

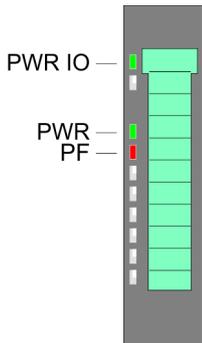
Valid address may range from 1 to 125. Addresses must be unique on the bus. The slave address must have been preset before the bus coupler is turned on.



Pos.	Value	Example	
		State	Address
1	not used	---	1+2+32=35 Address: 35
2	1	1	
3	2	1	
4	4	0	
5	8	0	
6	16	0	
7	32	1	
8	64	0	

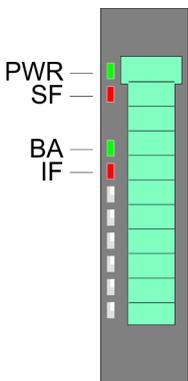
### 3.2.3 LEDs

#### LEDs power module



PWR IO	PWR	PF	Description
<span style="color: green;">■</span> green	<span style="color: green;">■</span> green	<span style="color: red;">■</span> red	
<span style="color: green;">■</span>	X	<input type="checkbox"/>	Power section supply OK
<span style="color: green;">■</span>	<span style="color: green;">■</span>	<input type="checkbox"/>	Electronic section supply OK
X	X	<span style="color: red;">■</span>	Fuse electronic section supply defective
not relevant: X			

#### Status indication Bus interface



LED	Description	
PWR <span style="color: green;">■</span> green	Bus interface is power supplied	
SF <span style="color: red;">■</span> red	Station fault, structure is not corresponding to the configuration	
BA <span style="color: green;">■</span> green	Operation mode: operational (ready for data exchange)	
	<span style="color: green;">▣</span> green 2Hz	Operation mode: pre-operational
	<span style="color: green;">▣</span> green 10Hz	Operation mode: prepared
IF <span style="color: red;">■</span> red	Internal error occurred	

For the fast diagnosis of the current module status 4 LEDs are on the front side. ↪ *Chap. 4.5 'LED status indication' page 64.*

### 3.3 Technical data

<b>Order no.</b>	<b>053-1CA00</b>
Type	IM 053CAN
Module ID	-
<b>Technical data power supply</b>	
Power supply (rated value)	DC 24 V
Power supply (permitted range)	DC 20.4...28.8 V
Reverse polarity protection	✓
Current consumption (no-load operation)	90 mA
Current consumption (rated value)	0.95 A
Inrush current	3.9 A
$I^2t$	0.14 A <sup>2</sup> s
Max. current drain at backplane bus	3 A
Max. current drain load supply	10 A
Power loss	3 W
<b>Status information, alarms, diagnostics</b>	
Status display	yes
Interrupts	yes, parameterizable
Process alarm	no
Diagnostic interrupt	yes, parameterizable
Diagnostic functions	yes, parameterizable
Diagnostics information read-out	possible
Supply voltage display	green LED
Service Indicator	-
Group error display	red LED
Channel error display	none
<b>Hardware configuration</b>	
Racks, max.	1
Modules per rack, max.	64
Number of digital modules, max.	64
Number of analog modules, max.	64
<b>Communication</b>	
Fieldbus	CANopen
Type of interface	CAN
Connector	Sub-D, 9-pin, male
Topology	Linear bus with bus termination at both ends
Electrically isolated	✓

<b>Order no.</b>	<b>053-1CA00</b>
Number of participants, max.	127
Node addresses	1 - 127
Transmission speed, min.	10 kbit/s
Transmission speed, max.	1 Mbit/s
Address range inputs, max.	128 Byte
Address range outputs, max.	128 Byte
Number of TxPDOs, max.	16
Number of RxPDOs, max.	16
<b>Datasizes</b>	
Input bytes	-
Output bytes	-
Parameter bytes	-
Diagnostic bytes	-
<b>Housing</b>	
Material	PPE / PPE GF10
Mounting	Profile rail 35 mm
<b>Mechanical data</b>	
Dimensions (WxHxD)	48.5 mm x 109 mm x 76.5 mm
Net weight	162.5 g
Weight including accessories	162.5 g
Gross weight	180 g
<b>Environmental conditions</b>	
Operating temperature	0 °C to 60 °C
Storage temperature	-25 °C to 70 °C
<b>Certifications</b>	
UL certification	yes
KC certification	yes

## 4 Deployment

### 4.1 Basics CAN

#### General

- CANopen (Control Area Network) is an international standard for open fieldbus systems intended for building, manufacturing and process automation applications that was originally designed for automotive applications.
- Due to its extensive error detection facilities, the CAN bus system is regarded as the most secure bus system. It has a residual error probability of less than  $4.7 \times 10^{-11}$ . Bad messages are flagged and retransmitted automatically.
- In contrast to PROFIBUS and INTERBUS, CAN defines under the CAL-level-7-protocol (CAL=CAN application layer) defines various level-7 user profiles for the CAN bus. One standard user profile defined by the CIA (CAN in Automation) e.V. is CANopen.

#### CANopen

- CANopen is a user profile for industrial real-time systems, which is currently supported by a large number of manufacturers. CANopen was published under the heading of DS-301 by the CAN in Automation association (CIA). The communication specifications DS-301 define standards for CAN devices. These specifications mean that the equipment supplied by different manufacturers is interchangeable. The compatibility of the equipment is further enhanced by the equipment specification DS-401 that defines standards for the technical data and process data of the equipment. DS-401 contains the standards for digital and analog input/output modules.
- CANopen comprises a communication profile that defines the objects that must be used for the transfer of certain data as well as the device profiles that specify the type of data that must be transferred by means of other objects.
- The CANopen communication profile is based upon an object directory that is similar to the profile used by PROFIBUS. The communication profile DS-301 defines two standard objects as well as a number of special objects:
  - Process data objects (PDO)  
PDOs are used for real-time data transfers
  - Service data objects (SDO)  
SDOs provide access to the object directory for read and write operations

#### EDS file

- For every CANopen slave from Yaskawa there is a EDS file available. This file may either be found on the supplied storage media or at the download area of [www.yaskawa.eu.com](http://www.yaskawa.eu.com).
- Please install the according file into your configuration tool. Details on the installation of the EDS file are available from the manual supplied with your configuration tool.

#### Communication medium

- CAN is based on a linear bus topology. You can use router nodes to construct a network. The number of devices per network is only limited by the performance of the bus driver modules.
- The maximum distance covered by the network is determined by the runtimes of the signals. This means that a data rate of 1Mbit/s limits the network to 40m and 80kbit/s limits the network to 1000m.
- The CAN bus communication medium employs a screened three-core cable (optionally a five-core). The CAN bus operates by means of differential voltages. For this reason it is less sensitive to external interference than a pure voltage or current based interface. The network must be configured as a serial bus, which is terminated by a  $120\Omega$  terminating resistor.

- Your CP contains a 9pin socket. You must use this socket to connect the CAN bus coupler as a slave directly to your CAN bus network.
- All devices on the network use the same transfer rate. Due to the bus structure of the network it is possible to connect or disconnect any station without interruption to the system. It is therefore also possible to commission a system in various stages. Extensions to the system do not affect the operational stations. Defective stations or new stations are recognized automatically.

### Bus access method

- Bus access methods are commonly divided into controlled (deterministic) and uncontrolled (random) bus access systems.
- CAN employs a Carrier-Sense Multiple Access (CSMA) method, i.e. all stations have the same right to access the bus as long as the bus is not in use (random bus access).
- Data communications is message related and not station related. Every message contains a unique identifier, which also defines the priority of the message. At any instance only one station can occupy the bus for a message.
- CAN-Bus access control is performed by means of a collision-free, bit-based arbitration algorithm. Collision-free means that the final winner of the arbitration process does not have to repeat his message. The station with the highest priority is selected automatically when more than one station accesses the bus simultaneously. Any station that is has information to send will delay the transmission if it detects that the bus is occupied.

## 4.2 Fast introduction

### Overview

This section is for experienced CANopen user that are already common with CAN. It will be shortly outlined, which messages are necessary for the deployment of the System SLIO with CAN in the start configuration.



Please regard that this manual prints the hexadecimal numbers in the type for developers "0x".  
E.g.: **0x15AE = 15AEh**

### Adjusting transfer rate and module-ID

Via the address selector you have to adjust a common transfer rate at the bus couplers as well as different node-IDs.

After starting your power supply, you program the transfer rate and the module-ID via 00 at the address selector within 20s. ↪ *Chap. 4.4 'Transfer rate and module-ID' page 62.*

### CAN identifier

The CAN identifier for the in-/output data of the System SLIO are generated from the node addresses (1...125):

Data type	Default CAN identifier	Data type	Default CAN identifier
digital inputs 1 ... 64bits	0x180 + Node address	digital outputs 1 ... 64bits	0x200 + Node address
analog inputs 1 ... 4words	0x280 + Node address	analog outputs 1 ... 4 Words / Channels	0x300 + Node address
other digital or analog inputs	0x380 + Node address	other digital or analog outputs	0x400 + Node address
	0x480 + Node address		0x500 + Node address
More identifiers are blocked per default and may be activated via SDO telegram.			

### Digital in-/outputs

The CAN messages with digital input data are represented as follows:

#### Identifier 0x180+Node address + up to 8byte user data

Identifier 11bit      DI 0 8bit      DI 1 8bit      DI 2 8bit      ...      DI 7 8bit

The CAN messages with digital output data are represented as follows:

#### Identifier 0x200+Node address + up to 8byte user data

Identifier 11bit      DO 0 8bit      DO 1 8bit      DO 2 8bit      ...      DO 7 bit

### Analog in-/outputs

The CAN messages with analog input data are represented as follows:

**Identifier 0x280+Node address + up to 4Words user data**

**Identifier** 11bit                      **AI 0** 1word                      **AI 1** 1word                      **AI 2** 1word                      **AI 3** 1word

The CAN messages with analog output data are represented as follows:

**Identifier 0x300+Node address + up to 4Words user data**

**Identifier** 11bit                      **AI 0** 1word                      **AI 1** 1word                      **AI 2** 1word                      **AI 3** 1word

**Network management  
NMT**

The network management controls the start-up of the network and the device monitoring. The NMT telegram has the following structure:

**Identifier 0x000+Command Specifier + Node address**

**Identifier** 11bit                      **CS** 8bit                      **ID** 8bit

Command specifier CS:

0x01: enter operational

0x02: enter prepared

0x80: enter pre-operational

0x81: reset node

0x82: reset communication

ID = 0: Broadcast to every node

ID = 1 ... 127: 127: Addressing the device with ID = Node address

**Node Guarding**

For the System SLIO works per default in event-controlled mode (no cyclic DataExchange), a node failure is not always immediately detected. Remedy is the control of the nodes per cyclic state request (Node Guarding). You request cyclically a state telegram via Remote-Transmit-Request (RTR): the telegram only consists of a 11bit identifier:

**Identifier 0x700+Node address**

**Identifier** 11bit

The node answers with a telegram that contains one state byte:

**Identifier 0x700+Node address + State byte**

**Identifier** 11bit                      **Status** 8bit

Bit 0 ... 6: Node state

0x7F: Pre-Operational

0x05: Operational

0x04: Stopped res. Prepared

Bit 7: Toggle bit, toggles after every send

To enable the bus coupler to recognize a network master failure (watchdog function), you still have to set the Guard-Time (Object 0x100C) and the Life-Time-Factor (Object 0x100D) to values≠0. (reaction time at failure: Guard-Time x Life Time Factor).

**Heartbeat**

Besides the Node Guarding, the IM 053-1CA00 also supports the Heartbeat Mode. If there is a value set in the index 0x1017 (Heartbeat Producer Time), the device state (Operational, Pre-Operational, ...) is transferred when the Heartbeat-Timer run out by using the COB identifier (0x700+Module-ID):

**Identifier 0x700+Node address + State byte**

**Identifier** 11bit                      **Status** 8bit

The Heartbeat Mode starts automatically as soon as there is a value in index 0x1017 higher 0.

**Emergency Object**

To send internal device failures to other participants at the CAN bus with a high priority, the IM 053-1CA00 supports the Emergency Object.

To activate the emergency telegram, you need the **COB-Identifier** Identifier that is fixed after boot-up in the object directory from the index 0x1014 in hexadecimal view: **0x80 + Modul-ID**.

The emergency telegram has always a length of 8byte. It consists of:

**Identifier 0x80 + Node address + 8byte user data**

**Identifier** 11bit              **EC0**    **EC1**    **Ereg**    **Inf0**    **Inf1**    **Inf2**    **Inf3**    **Inf4**

**Error messages**

An overview of the error messages is to be found further below at "Emergency Object".



*The now described telegrams enable you to start and stop the System SLIO, read inputs, write outputs and control the modules. In the following, the functions are described in detail.*

## 4.3 Accessing the System SLIO

### 4.3.1 General

#### Overview

In the following you will find the description of accessing the following System SLIO areas via CAN:

- I/O area
- Parameter data
- Diagnostics data

Information concerning the allocation of these areas may be found in the description of the corresponding System SLIO module.

#### EDS file

To configure the slave connections in your own configuration tool, you've got all the information about your modules in form of an electronic data sheet file. Install this EDS file in your configuration tool. The current EDS file may be found in the download area of [www.yaskawa.eu.com](http://www.yaskawa.eu.com). More information about installing the EDS may be found at the online help of the according engineering tool.

### 4.3.2 Accessing I/O area

The I/O area may be accessed via the following objects:

- PDO
- SDO

#### Access via PDO

In PowerON the CAN coupler creates a default configuration (see below).

As soon as the CAN coupler is set to operational with the NMT command (01h: "enter operational"), the input data of the digital, analog and function modules are transferred once via TxPDO.

In the default configuration the TxPDOs of every module with exception of the analog modules are set to event controlled. This means as soon as data are changed, these were transferred via TxPDO.

With the analog inputs per default there is no data transfer at change of value. This is to avoid overloading the bus. By writing a value > 0 to index 0x6423/subindex 0 in the object directory the module is triggered to send PDOs at change value. With writing 0 to this index this function is again deactivated.

The output data are written via RxPDOs and directly sent to the outputs. Here PDO length is checked by the CAN coupler.

#### **Mapping sequence of the module classes**

1. Digital modules
  2. Analog modules
  3. CPs
  4. Function modules: counter
  5. Function modules: SSI
  6. Function modules: PWM
  7. Function modules: ETS
- (per default with each module 6 ETS entries are mapped)

**Default configuration**

Identifier (CobIDs) of the TxPDOs for the inputs		
PDO no.	Mapping of the inputs	Identifier
1	Digital 1-8	0x180
2	Analog 1-4	0x280
3	Digital 9-16, Analog 5-8, CP, Function module	0x380
4	Digital 17-24, Analog 9-12, CP, Function module	0x480
5	Digital 17-32, Analog 13-16, CP, Function module	*
6	Digital 33-40, Analog 17-20, CP, Function module	*
7	Digital 41-48, Analog 21-24, CP, Function module	*
8	Digital 49-56, Analog 25-28, CP, Function module	*
9	Digital 57-64, Analog 29-32, CP, Function module	*
10	Analog 33-36, CP, Function module	*
11	Analog 37-40, CP, Function module	*
12	Analog 41-44, CP, Function module	*
13	Analog 45-48, CP, Function module	*
14	Analog 49-52, CP, Function module	*
15	Analog 53-56, CP, Function module	*
16	Analog 57-60, CP, Function module	*

Identifier (CobIDs) of the RxPDOs for the outputs		
PDO no.	Mapping	Identifier
1	Digital 1-8	0x200
2	Analog 1-4	0x300
3	Digital 9-16, Analog 5-8, CP, Function module	0x400
4	Digital 17-24, Analog 9-12, CP, Function module	0x500
5	Digital 17-32, Analog 13-16, CP, Function module	*
6	Digital 33-40, Analog 17-20, CP, Function module	*
7	Digital 41-48, Analog 21-24, CP, Function module	*
8	Digital 49-56, Analog 25-28, CP, Function module	*
9	Digital 57-64, Analog 29-32, CP, Function module	*
10	Analog 33-36, CP, Function module	*
11	Analog 37-40, CP, Function module	*
12	Analog 41-44, CP, Function module	*
13	Analog 45-48, CP, Function module	*
14	Analog 49-52, CP, Function module	*
15	Analog 53-56, CP, Function module	*
16	Analog 57-60, CP, Function module	*

\*) PDO is per default blocked and may be accordingly set and activated with PDO-Linking.

**Access via SDO**

The object directory may be r/w accessed via SDO (service data object). Here data of any length may be transferred. If necessary the data were distributed to several CAN messages with the same identifier (segmentation). Transfers by means of SDO are always with acknowledgement of the partner.

**Accessing input data**

SDO Upload Request for input data:

**Identifier 0x600+Node address + up to 8byte request data for input data**

<b>Identifier</b>	<b>0x40</b>	<b>Index Low</b>	<b>Index Low</b>	<b>Subindex</b>	<b>0x00</b>	<b>0x00</b>	<b>0x00</b>	<b>0x00</b>
11bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit

SDO Upload Response with input data:

**Identifier 0x580+Node address + up to 8byte input data**

<b>Identifier</b>	<b>0x4F</b>	<b>Index Low</b>	<b>Index High</b>	<b>Subindex</b>	<b>Data 0</b>	<b>Data 1</b>	<b>Data 2</b>	<b>Data 3</b>
11bit	8Bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit
	<b>0x4B</b>							
	16bit							
	<b>0x43</b>							
	32bit							

**Writing output data**

SDO Download Request for output data:

**Identifier 0x600+Node address + up to 8byte output data**

<b>Identifier</b>	<b>0x2F</b>	<b>Index Low</b>	<b>Index High</b>	<b>Subindex</b>	<b>Data 0</b>	<b>Data 1</b>	<b>Data 2</b>	<b>Data 3</b>
11bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit
	<b>0x2B</b>							
	16bit							
	<b>0x23</b>							
	32bit							

SDO Download Response with response data:

**Identifier 0x580+Node address + up to 8byte response data**

<b>Identifier</b>	<b>0x60</b>	<b>Index Low</b>	<b>Index High</b>	<b>Subindex</b>	<b>0x00</b>	<b>0x00</b>	<b>0x00</b>	<b>0x00</b>
11bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit



*Instead of the 4byte with 0x00, SDO error codes may be returned. More may be found at "SDO".*

### 4.3.3 Accessing parameter data

The modules are parameterized by SDO transfer. Here a parameterizable module is addressed by subindex at the system bus. The parameters may be accessed by indices. Information concerning the index allocation may be found in the description of the corresponding System SLIO module. It is valid:

- Subindex 0x01: 1. parameterizable module at the bus
  - Subindex 0x02: 2. parameterizable module at the bus
- ect.
- Index 0x3100: 1. parameter byte
  - Index 0x3101: 2. parameter byte
- ect.

#### Write parameter data

SDO Download Request for parameter data:

#### Identifier 0x600+Node address + up to 8byte output data

Identifier	0x2F	Index Low	Index High	Subindex	Data 0	Data 1	Data 2	Data 3
11bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit

SDO Download Response with response data:

#### Identifier 0x580+Node address + up to 8byte response data

Identifier	0x60	Index Low	Index High	Subindex	0x00	0x00	0x00	0x00
11bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit

#### Set parameter data

With a write access to index 31FFh and subindex 0, the parameters of each module at one bus coupler are set.

SDO Download Request to set parameters:

#### Identifier 0x600+Node address + up to 8byte data

Identifier	0x2F	0xFF	0x31	0x00	0x01	0x00	0x00	0x00
11bit	8bit							

SDO Download Response with response data:

#### Identifier 0x580+Node address + up to 8byte response data

Identifier	0x60	0xFF	0x31	0x00	0x00	0x00	0x00	0x00
11bit	8bit							



*Instead of the 4byte with 0x00, SDO error codes may be returned. More may be found at "SDO".*

*↳ Chap. 4.8 'SDO' page 70*

### 4.3.4 Accessing diagnostics data

Process and diagnostic interrupt data of System SLIO modules with interrupt capability were automatically sent by an emergency telegram if the interrupt is activated by parameterization. There is the possibility to request diagnostics data via SDO.

#### Emergency-Telegram

In the emergency telegram the first 4byte of the process respectively diagnostics data may be found. The emergency telegram has the following structure:

#### Process interrupt data

**Identifier 0x80+Node address + up to 8byte response data**

Identifier	0x00	0x00	0x00	0x80	Data 0	Data 1	Data 2	Data 3
11bit	8bit	8bit	8bit	+Slot no. 8bit	8bit	bit	8bit	8bit

#### Diagnostics interrupt data

**Identifier 0x80+Node address + up to 8byte response data**

Identifier	0x00	0x00	0x00	0x40	Data 0	Data 1	Data 2	Data 3
11bit	8bit	8bit	8bit	+Slot no. 8bit	8bit	8bit	8bit	8bit

#### Access via SDO

The diagnostics data may be accessed via SDO. Here the segmented SDO protocol is used. More about the segmented SDO protocol may be found at [www.can-cia.org](http://www.can-cia.org) at CiA301.

Information concerning the index allocation may be found in the description of the corresponding System SLIO module. With subindex the module slot starting with 1 is defined.

SDO Upload Request for diagnostics data:

**Identifier 0x600+Node address + up to 8byte request data for diagnostics**

Identifier	0x40	Index Low	Index High	Subindex	0x00	0x00	0x00	0x00
11bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit

SDO Upload Response with diagnostics data:

**Identifier 0x580+Node address + up to 8byte input data**

Identifier	0x41	Index Low	Index High	Subindex	Data 0	Data 1	Data 2	Data 3
11bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit	8bit

#### Bus recording segmented SDO protocol

Slot: 1, Node address: 2:

```
0602 8 40 01 2F 01 00 00 00 00
0582 8 41 01 2F 01 14 00 00 00
0602 8 60 00 00 00 00 00 00 00
0582 8 00 00 1F 00 00 70 00 08
0602 8 70 00 00 00 00 00 00 00
```

```
0582 8 10 00 00 00 00 00 00 00
0602 8 60 00 00 00 00 00 00 00
0582 8 03 00 00 00 00 07 C7 00
```

### 4.4 Transfer rate and module-ID

**Overview**

There is the possibility to specify transfer rate and module-ID by means of the address selector. The settings are permanently stored in an EEPROM and may any time be changed. Per default the bus coupler has a transfer rate of 500kbit/s.

**Proceeding**

1. Turn off the power supply of the bus coupler
2. Set the address selector to 0.
3. Turn on the power supply of the bus coupler. The LEDs SF, BA and IF are blinking. Now for each there is a period of 10s to select *transfer rate* and *module-ID*.

**i** If the address selector is still 0 after 20s, the following values are set:

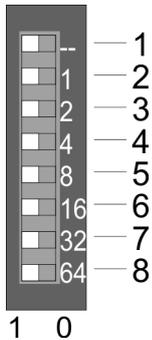
- Transfer rate: 1Mbit/s
- Module-ID: 0 (not valid)

To change these settings start with the "Proceeding" again.

**Programming transfer rate**

As long as the LEDs SF, BA and IF are blinking, select the transfer rate with the address selector.

There are the following possibilities for setting:



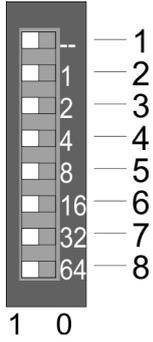
Pos.	Value	Transfer rate kbit/s									
		1000	800	500	250	125	100	50	20	10	
1	---	---	---	---	---	---	---	---	---	---	---
2	1	0	0	1	0	1	0	1	0	1	
3	2	0	0	0	1	1	0	0	1	1	
4	4	0	0	0	0	0	1	1	1	1	
5	8	0	1	0	0	0	0	0	0	0	
6	16	0	0	0	0	0	0	0	0	0	
7	32	0	0	0	0	0	0	0	0	0	
8	64	0	0	0	0	0	0	0	0	0	
		<b>25</b>	<b>50</b>	<b>100</b>	<b>250</b>	<b>500</b>	<b>600</b>	<b>1000</b>	<b>2500</b>	<b>5000</b>	
		<b>max. guaranteed bus distance in m</b>									

After 10s the selected transfer rate is stored in the EEPROM and the IF LED gets off.

**Programming module-ID**

For a further period of 10s, as long as the LEDs SF and BA are blinking, you may set the module-ID in a range of 1 ... 125 by means of the address selector.

There are the following possibilities for setting:



Pos.	Value	Example State	Address
1	---	---	1+2+32=35
2	1	1	Address: 35
3	2	1	
4	4	0	
5	8	0	
6	16	0	
7	32	1	
8	64	0	

After 10s the selected module-ID is taken and the bus coupler returns to the normal operating mode (status: "Pre-Operational") with the set values.

 *Each module-ID may only exist once at the bus! The module-ID must be set before the bus coupler is powered on!*

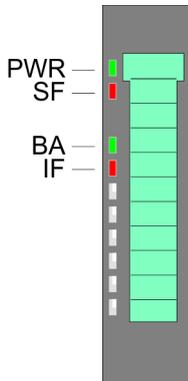
**Transfer rate selection by SDO-write**

You can also modify the CAN transfer rate by means of an SDO-Write operation to the object "0x2001". The entered value is used as the CAN transfer rate when the bus coupler has been RESET. This method is a most convenient when you must change the CAN transfer rate of all the bus couplers of a system from a central CAN terminal. The bus couplers use the programmed transfer rate when the system has been RESET.

### 4.5 LED status indication

**General**

The LEDs installed to display the status allow extensive diagnostics during the PowerON procedure as well as during operation. The result of the diagnosis is determined by the combination of the different LEDs and the current operating mode.



PWR ■ green	SF ■ red	BA ■ green	IF ■ red	Status
□	□	□	□	There is no power supply or the power module is defective.
■	X	X	X	The bus interface is power supplied.
■	X	□	X	Phase of initialization
■	X	▣ 10Hz	X	Operating mode: Prepared
■	X	▣ 2Hz	X	Operating mode: Pre-operational
■	X	■	X	Operating mode: Operational
■	▣ 2Hz	▣ 2Hz	▣ 2Hz	Setting of the transfer rate is activated. With the address switch set the transfer rate within the next 10s.
■	▣ 2Hz	▣ 2Hz	□	Setting of the module ID is activated. With the address switch set the module ID within the next 10s.
■	▣ 2Hz	□	▣ 2Hz	Error in the set transfer rate. Adjust it again.
■	■	□	▣ 2Hz	Error in the set module ID. Adjust it again.
■	□	□	▣ 2Hz	Internal error on the CAN bus coupler.
■	■	X	■	Error on the System SLIO bus, please check your modules.
■	■	▣ 2Hz	▣ 2Hz	Operating mode: pre-operational with an internal error on a System SLIO module.
■	■	▣ 2Hz	□	Operating mode: pre-operational with configuration error on System SLIO bus (see index 0x2F02 and 0x2010).
■	■	X	□	Error in project engineering or there is a diagnostic pending.

not relevant: X

## 4.6 Telegram structure

**Identifier** The CANopen telegrams have the following structure according to CiA DS-301:

### Identifier

Byte	Bit 7 ... Bit 0
1	Bit 3 ... Bit 0: most significant 4 bits of the module-ID Bit 7 ... Bit 4: CANopen function code
2	Bit 3 ... Bit 0: data length code (DLC) Bit 4: RTR bit: 0: no data (request code) 1: data available Bit 7 ... Bit 5: Least significant 3 bits of the module-ID

### Data

### Data

Byte	Bit 7 ... Bit 0
3 ... 10	Data

An additional division of the 2 byte identifier into function portion and a module-ID gives the difference between this and a level 2 telegram. The function determines the type of telegram (object) and the module-ID addresses the receiver. CANopen devices exchange data in the form of objects. The CANopen communication profile defines two different object types as well as a number of special objects.

The SLIO CAN bus coupler supports the following objects:

- 16 transmit PDOs (PDO Linking, PDO Mapping)
- 16 receive PDOs (PDO Linking, PDO Mapping)
- 2 standard SDOs
- 1 emergency object
- 1 network management object NMT
- Node Guarding
- Heartbeat

**CANopen function codes** Every object is associated with a function code. You can obtain the required function code from the following table:

Object	Function code (4 bits)	Receiver	Definition	Function
NMT	0000	Broadcast	CiA301	Network management
EMERGENCY	0001	Master	CiA301	Error telegram
PDO1S2M	0011	Master, Slave (RTR)	CiA301	Digital input data 1
PDO1M2S	0100	Slave	CiA301	Digital output data 1
SDO1S2M	1011	Master	CiA301	Configuration data
SDO1M2S	1011	Slave	CiA301	Configuration data

Telegram structure

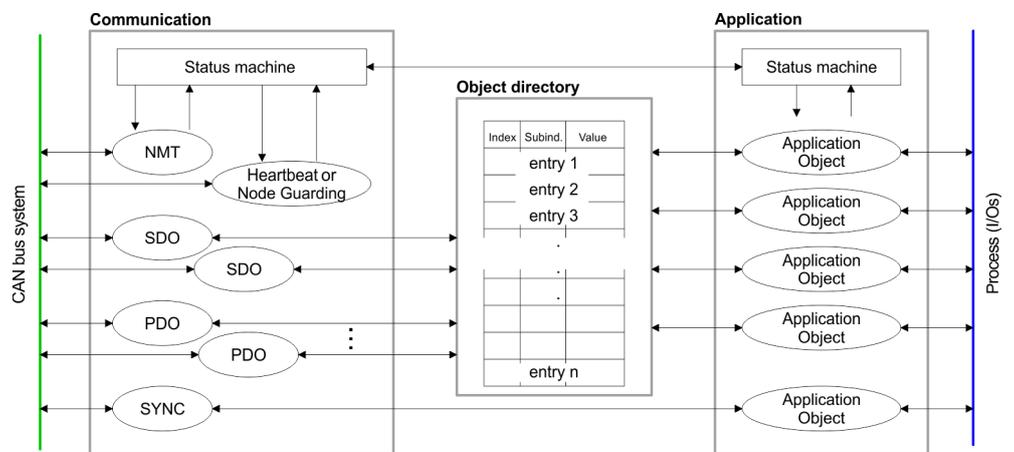
Object	Function code (4 bits)	Receiver	Definition	Function
Node Guarding	1110	Master, Slave (RTR)	CiA301	Module monitoring
Heartbeat	1110	Master, Slave	Application spec.	Module monitoring



A detailed description of the structure and the contents of these objects is available in "CiA Communication Profile DS-301 Version 3.0" and "CiA Device Profile for I/O-Modules CiA301, CiA401", which can be found at [www.can-cia.org](http://www.can-cia.org).

Structure of the device model

A CANopen device can be structured as follows:



Communication

Serves the communication data objects and the concerning functionality for data transfer via the CANopen network.

Application

The application data objects contain e.g. in- and output data. In case of an error, an application status machine switches the outputs in a secure state. The object directory is organized as 2 dimension table. The data is addressed via index and sub-index.

Object directory

This object directory contains all data objects (application data + parameters) that are accessible and that influence the behavior of communication, application and status machines.

## 4.7 PDO

### 4.7.1 PDO overview

#### General

In many fieldbus systems the whole process image is transferred - mostly more or less cyclically. CANopen is not limited to this communication principle, for CAN supports more possibilities through multi master bus access coordination. CANopen divides the process data into segments of max. 8byte. These segments are called process data objects (PDOs). Every PDO represents one CAN telegram and is identified and prioritized via its specific CAN identifier.

For the exchange of process data, the System SLIO CAN bus coupler supports 32 PDOs. Every PDO consists of a maximum of 8 data bytes. The transfer of PDOs is not verified by means of acknowledgments since the CAN protocol guarantees the transfer.

There are 16 Tx transmit PDOs for input data and 16 Rx receive PDOs for output data. The PDOs are named seen from the bus coupler:

- Receive PDOs (RxPDOs) are received by the bus coupler and contain output data.
- Transmit PDOs (TxPDOs) are send by the bus coupler and contain input data.

The assignment of the PDOs to input or output data occurs automatically.

#### Variable PDO mapping

CANopen predefines the first two PDOs in the device profile. The assignment of the PDOs is fixed in the mapping tables in the object directory. The mapping tables are the cross-reference between the application data in the object directory and the sequence in the PDOs.

The assignment of the PDOs, automatically created by the coupler (Default configuration), are commonly adequate. For special applications, the assignment may be changed. Herefore you have to configure the mapping tables accordingly. First, you write a 0 to sub-index 0 (deactivates the current mapping configuration). Then you insert the wanted application objects into sub-index 1...8. Finally you parameterize the number of now valid entries in sub-index 0 and the coupler checks the entries for their consistency.

#### Mapping sequence of the module classes

1. Digital modules
  2. Analog modules
  3. CPs
  4. Function modules: counter
  5. Function modules: SSI
  6. Function modules: PWM
  7. Function modules: ETS
- (per default with each module 6 ETS entries are mapped)

#### Default configuration

Identifier (CobIDs) of the TxPDOs for the inputs		
PDO no.	Mapping of the inputs	Identifier
1	Digital 1-8	0x180
2	Analog 1-4	0x280
3	Digital 9-16, Analog 5-8, CP, Function module	0x380
4	Digital 17-24, Analog 9-12, CP, Function module	0x480
5	Digital 17-32, Analog 13-16, CP, Function module	*
6	Digital 33-40, Analog 17-20, CP, Function module	*
7	Digital 41-48, Analog 21-24, CP, Function module	*

**Identifier (CobIDs) of the TxPDOs for the inputs**

PDO no.	Mapping of the inputs	Identifier
8	Digital 49-56, Analog 25-28, CP, Function module	*
9	Digital 57-64, Analog 29-32, CP, Function module	*
10	Analog 33-36, CP, Function module	*
11	Analog 37-40, CP, Function module	*
12	Analog 41-44, CP, Function module	*
13	Analog 45-48, CP, Function module	*
14	Analog 49-52, CP, Function module	*
15	Analog 53-56, CP, Function module	*
16	Analog 57-60, CP, Function module	*

**Identifier (CobIDs) of the RxPDOs for the outputs**

PDO no.	Mapping	Identifier
1	Digital 1-8	0x200
2	Analog 1-4	0x300
3	Digital 9-16, Analog 5-8, CP, Function module	0x400
4	Digital 17-24, Analog 9-12, CP, Function module	0x500
5	Digital 17-32, Analog 13-16, CP, Function module	*
6	Digital 33-40, Analog 17-20, CP, Function module	*
7	Digital 41-48, Analog 21-24, CP, Function module	*
8	Digital 49-56, Analog 25-28, CP, Function module	*
9	Digital 57-64, Analog 29-32, CP, Function module	*
10	Analog 33-36, CP, Function module	*
11	Analog 37-40, CP, Function module	*
12	Analog 41-44, CP, Function module	*
13	Analog 45-48, CP, Function module	*
14	Analog 49-52, CP, Function module	*
15	Analog 53-56, CP, Function module	*
16	Analog 57-60, CP, Function module	*

\* PDO is per default blocked and may be accordingly set and activated with PDO-Linking.

**PDO identifier COB-ID**

The most important communication parameter of a PDOs is the CAN identifier (also called "Communication Object Identifier", COB-ID). It serves the identification of the data and sets the priority of bus access. For every CAN data telegram only one sending node may exist (producer). Due to the ability of CAN to send all messages per broadcast procedure, however, a telegram may be received by several bus participants at the same time (consumer). Therefore, one node may deliver its input information to different bus stations similarly - without needing the pass through a logical bus master.

The System SLIO provides receive and transmit PDOs default identifier in dependence of the node address. Below follows a list of the COB identifiers for the receive and the transmit PDO transfer that are pre-set after boot-up. The transmission type in the object directory (indices 0x1400-0x140F and 0x1800-0x180F, sub-index 0x02) is preset to asynchronous, event controlled (= 0xFF). The EVENT-timer (value \* 1ms) can be used to transmit the PDOs cyclically.

*Send:*

0x180 + Modul-ID: PDO1S2M digital (acc. DS-301)

0x280 + Modul-ID: PDO2S2M analog

0x380 + Modul-ID: PDO3S2M digital or analog

0x480 + Modul-ID: PDO4S2M

*Receive:*

0x200 + Modul-ID: PDO1M2S digital (acc. DS-301)

0x300 + Modul-ID: PDO2M2S analog

0x400 + Modul-ID: PDO3M2S digital or analog

0x500 + Modul-ID: PDO4M2S



*The PDOs 5 ... 16 are blocked per default and must be enabled by SDO telegrams. More information concerning this may be found in the object directory 1400 ... 1410 / 1800 ... 1810.*

## PDO linking

If the Consumer-Producer model of the CANopen PDOs shall be used for direct data transfer between nodes (without master), you have to adjust the identifier distribution accordingly, so that the TxPDO identifier of the producer is identical with the RxPDO identifier of the consumer:

This procedure is called PDO linking. this enables for example the simple installation of electronic gearing where several slave axis are listening to the actual value in TxPDO of the master axis.

## 4.7.2 PDO Communication types

CANopen supports the following possibilities for the process data transfer:

- Event triggered
- Polled
- Synchronized

### Event triggered

The "event" is the alteration of an input value, the data is send immediately after value change. The event control makes the best use of the bus width for not the whole process image is send but only the changed values. At the same time, a short reaction time is achieved, because there is no need to wait for a master request.

### Polled

PDOs may also be polled via data request telegrams (remote frames) to give you the opportunity to e.g. send the input process image of event triggered inputs to the bus without input change for example a monitoring or diagnosis device included during run-time. The SLIO CANopen bus coupler supports the query of PDOs via remote frames - for this can, due to the hardware, not be granted for all CANopen devices, this communication type is only partially recommended.

**Synchronized**

It is not only convenient for drive applications to synchronize the input information request and the output setting. For this purpose, CANopen provides the SYNC object, a CAN telegram with high priority and no user data which receipt is used by the synchronized nodes as trigger for reading of the inputs res. writing of the outputs.

**4.7.3 PDO transmission type**

The parameter "PDO transmission type" fixes how the sending of the PDOs is initialized and what to do with received ones:

Transmission Type	Cyclical	Acyclical	Synchronous	Asynchronous
0		x	x	
1-240	x		x	
254, 255				x

**Synchronous**

The transmission type 0 is only wise for RxPDOs: the PDO is analyzed at receipt of the next SYNC telegram. At transmission type 1-240, the PDO is send res. expected cyclically: after every "nth" SYNC ( $n = 1 \dots 240$ ). For the transmission type may not only be combined within the network but also with a bus, you may thus e.g. adjust a fast cycle for digital inputs ( $n = 1$ ), while data of the analog inputs is transferred in a slower cycle (e.g.  $n = 10$ ). The cycle time (SYNC rate) may be monitored (Object 0x1006), at SYNC failure, the coupler sets its outputs in error state.

**Asynchronous**

The transmission types 254 + 255 are asynchronous or also event triggered. The transmission type 254 provides an event defined by the manufacturer, at 255 it is fixed by the device profile.

When choosing the event triggered PDO communication you should keep in mind that in certain circumstances there may occur a lot of events similarly. This may cause according delay times for sending PDOs with lower priority values. You should also avoid to block the bus by assigning a high PDO priority to an often alternating input ("babbling idiot").

**Inhibit time**

Via the parameter "inhibit time" a "send filter" may be activated that does not lengthen the reaction time of the relatively first input alteration but that is active for the following changes. The inhibit time (send delay time) describes the min. time span that has to pass between the sending of two identical telegrams. When you use the inhibit time, you may ascertain the max. bus load and for this the latent time in the "worst case".

**4.8 SDO**

The Service Data Object (SDO) serves the read or write access to the object directory. The CAL layer 7 protocol gives you the specification of the Multiplexed-Domain-Transfer-Protocol that is used by the SDOs. This protocol allows you to transfer data of any length because where appropriate, messages are distributed to several CAN messages with the same identifier (segment building).

The first CAN message of the SDO contain process information in 4 of the 8bytes. For access to object directory entries with up to 4byte length, one single CAN message is sufficient. The following segments of the SDO contain up to 7byte user data. The last byte contains an end sign. A SDO is delivered with acknowledgement, i.e. every reception of a message is receipted.

The COB identifiers for read and write access are:

- Receive-SDO1: 0x600 + Modul-ID
- Transmit-SDO1: 0x580 + Modul-ID



*A detailed description of the SDO telegrams is to find in the DS-301 norm from CiA. In the following only the error messages are described that are generated at wrong parameterization.*

## SDO error codes

Code	Error
0x05030000	Toggle bit not alternated
0x05040000	SDO protocol timed out
0x05040001	Client/server command specifier not valid or unknown
0x05040002	Invalid block size (block mode only)
0x05040003	Invalid sequence number (block mode only)
0x05040004	CRC error (block mode only)
0x05040005	Out of memory
0x06010000	Unsupported access to an object
0x06010001	Attempt to read a write only object
0x06010002	Attempt to write a read only object
0x06020000	Object does not exist in the object dictionary
0x06040041	Object cannot be mapped to the PDO
0x06040042	The number and length of the objects to be mapped would exceed PDO length
0x06040043	General parameter incompatibility reason
0x06040047	General internal incompatibility in the device
0x06060000	Access failed due to an hardware error
0x06070010	Data type does not match, length of service parameter does not match
0x06070012	Data type does not match, length of service parameter too high
0x06070013	Data type does not match, length of service parameter too low
0x06090011	Sub-index does not exist
0x06090030	Value range of parameter exceeded (only for write access)
0x06090031	Value of parameter written too high
0x06090032	Value of parameter written too low
0x06090036	Maximum value is less than minimum value
0x08000000	general error
0x08000020	Data cannot be transferred or stored to the application
0x08000021	Data cannot be transferred or stored to the application because of local control

---

SDO

Code	Error
0x08000022	Data cannot be transferred or stored to the application because of the present device state
0x08000023	Object directory dynamic generation fails or no object directory is present (e.g. object directory is generated from file and generation fails because of an file error)

## 4.9 Object directory

### Structure

The CANopen object directory contains all relevant CANopen objects for the bus coupler. Every entry in the object directory is marked by a 16bit index.

If an object exists of several components (e.g. object type Array or Record), the components are marked via an 8bit sub-index.

The object name describes its function. The data type attribute specifies the data type of the entry.

The access attribute defines, if the entry may only be read, only be written or read and written.

The object directory is divided into the following 3 parts:

- **Communication specific profile area (0x1000 – 0x1FFF)**

This area contains the description of all relevant parameters for the communication.

0x1000 – 0x1011	General communication specific parameters (e.g. device name)
0x1400 – 0x140F	Communication parameters (e.g. identifier) of the receive PDOs
0x1600 – 0x160F	Mapping parameters of the receive PDOs The mapping parameters contain the cross-references to the application objects that are mapped into the PDOs and the data width of the depending object.
0x1800 – 0x180F 0x1A00 – 0x1A0F	Communication and mapping parameters of the transmit PDOs

- **Manufacturer specific profile area (0x2000 – 0x5FFF)**

Here you may find the manufacturer specific entries like e.g. PDO Control, CAN transfer rate (transfer rate after RESET) etc.

- **Standardized device profile area (0x6000 – 0x9FFF)**

This area contains the objects for the device profile acc. DS-401.



*For the CiA norms are exclusively available in English, we adapted the object tables. Some entries are described below the according tables.*

### Object directory overview

Index	Content of Object
0x1000	Device type
0x1001	Error register
0x1003	Error store
0x1004	Number of PDOs
0x1005	SYNC identifier
0x1006	SYNC interval
0x1007	Synchronous Window Length
0x1008	Device name
0x1009	Hardware version
0x100A	Software version

Index	Content of Object
0x100B	Node number
0x100C	Guard time
0x100D	Life time factor
0x100E	Node Guarding Identifier
0x1010	Save parameter
0x1011	Load parameter
0x1014	Emergency COB-ID
0x1016	Heartbeat consumer time
0x1017	Heartbeat producer time
0x1018	Device identification
0x1020	Verify Configuration - Date/Time
0x1027	Module list
0x1029	Error behavior
0x1400 - 0x140F	Communication parameter for Receive-PDOs (RxPDO, Master to Slave)
0x1600 - 0x160F	Mapping parameter for Receive-PDOs (RxPDO)
0x1800 - 0x180F	Communication parameter for Transmit-PDOs (TxPDO, Slave to Master)
0x1A00 - 0x1A0F	Mapping parameter for Transmit-PDOs (TxPDO)
0x2001	CAN transfer rate
0x2006	Sync-Cycle Jitter
0x200A	Software Package Version
0x200B	SLIO-Bus Version
0x2010	Save/Restore status
0x2028	Module: Device name
0x2029	Module: Hardware revision
0x202A	Module: Software revision
0x202B	Module: Serial number
0x2030	Can coupler: FPGA version
0x2031	Module: FPGA version
0x2040	Can coupler: MxFile
0x2041	Module: MxFile
0x2050	Can coupler: Product version
0x2051	Module: Product version
0x2400	PDO Control
0x2F00	Module: diagnostic data record 0
0x2F01	Module: diagnostic data record 1

Index	Content of Object
0x2F02	Module: initialization error
0x3100 - 0x31FF	Module Parameterization
0x3200	Access to record set bus coupler
0x3201 - 0x3240	Access to record set function modules
0x5000	Digital-Input-2/4-Bit Array (see DS 401)
0x5002	Polarity Digital-Input-2/4-Bit Array (see DS 401)
0x5200	Digital-Output-2/4-Bit Array (see DS 401)
0x5202	Polarity Digital-Output-2/4-Bit Array (see DS 401)
0x5206	Fault Mode Digital-Output-2/4-Bit Array (see DS 401)
0x5207	Fault State Digital-Output-2/4-Bit Array (see DS 401)

### Device Type

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1000	0	DeviceType	Unsigned32	ro	N	0x00050191	Statement of device type

The 32bit value is divided into two 16bit fields:

MSB	LSB
<b>Additional information Device</b>	<b>profile number</b>
0000 0000 0000 wxyz (bit)	401dec=0x0191

The "additional information" contains data related to the signal types of the I/O device:

z=1 → digital inputs

y=1 → digital outputs

x=1 → digital outputs

w=1 → analog outputs

### Error register

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1001	0	Error Register	Unsigned8	ro	Y	0x00	Error register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ManSpec	reserved	reserved	Comm.	reserved	reserved	reserved	Generic

- **ManSpec:**
  - Manufacturer specific error, specified in object 0x1003
- **Comm.:**
  - Communication error (overrun CAN)
- **Generic:**
  - A not more precisely specified error occurred (flag is set at every error message)

**Error store**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1003	0	Predefined error field (error store)	Unsigned8	ro	N	0x00	Object 0x1003 contains a description of the error that has occurred in the device - subindex 0 has the number of error states stored
	1	Actual error	Unsigned32	ro	N	Error code 1	Last error state to have occurred
	...	...	...	...	...	...	...
	254		Unsigned32	ro	N	Error code 254	A maximum of 254 error states

The "predefined error field" is divided into two 16bit fields:

MSB	LSB
Additional information	Error code

- Error code and Additional information are described at the table "Error messages" from Emergency Object.
- New errors are always saved at subindex 1, all the other subindices being appropriately incremented.
- By writing a "0" to subindex 0, the whole error memory is cleared. If there has not been an error since PowerOn, then object 0x1003 exists only of subindex 0 with entry "0".
- Via reset or PowerCycle, the error memory is cleared.

**Error codes**

Meaning	Byte 4	Byte 3	Byte 2	Byte 1
	Bit 31...24	Bit 32...16	Bit 15...8	Bit 7...0
No error	0x00 (fix)	0x00 (fix)	0x00 (fix)	0x00 (fix)
CANopen initialization failure	0x01 (fix)	0x00 (fix)	0x10 (fix)	0x00 (fix)
Sync cycle error	0x00 (fix)	0x00 (fix)	0x81 (fix)	0x01 (fix)
Lifeguard error or Heartbeat error	0x00 (fix)	0x00 (fix)	0x81 (fix)	0x30 (fix)
Slice bus error	Slice bus module number (range: 0..63)	0x00 (fix)	0x81 (fix)	0x57 (fix)
PDO not processed due to length error	PDO length	PDO number	0x82 (fix)	0x10 (fix)
PDO length exceeded	PDO length	PDO number	0x82 (fix)	0x20 (fix)

Meaning	Byte 4	Byte 3	Byte 2	Byte 1
	Bit 31...24	Bit 32...16	Bit 15...8	Bit 7...0
PDO TX mapping error	PDO number	0x16 (fix)	0x63 (fix)	0x00 (fix)
PDO RX mapping error	PDO number	0x1A (fix)	0x63 (fix)	0x00 (fix)
Slice bus diagnostic alarm	0x40 (fix)	Slice bus module number (range: 1..64)	0x10 (fix)	Diagnostic byte 0
Slice bus process alarm	0x80 (fix)	Slice bus module number (range: 1..64)	0x00 (fix)	0x00 (fix)
Slice bus initialization error	0xE0 (fix)	0x00 (fix)	0x10 (fix)	0x00 (fix)
Slice bus communication error	0xE1 (fix)	0x00 (fix)	0x10 (fix)	0x00 (fix)
Slice bus CP module I/O size > 8byte	0xE2 (fix)	0x00 (fix)	0x20 (fix)	0x00 (fix)
Slice bus module error	0xE3 (fix)	Slice bus module number (range: 1..64)	0x10 (fix)	0x00 (fix)

### Number of PDOs

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1004	0	Number of PDOs supported	Unsigned32	ro	N	0x00100010	Number of PDOs supported
	1	Number of synchronous PDOs supported	Unsigned32	ro	N	0x00100010	Number of synchronous PDOs supported
	2	Number of asynchronous PDOs supported	Unsigned32	ro	N	0x00100010	Number of asynchronous PDOs supported

The 32bit value is divided into two 16bit fields:

MSB	LSB
Number of receive (Rx)PDOs supported	Number of send (Tx)PDOs supported

### SYNC identifier

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1005	0	COB-Id syncmessage	Unsigned32	ro	N	0x00000080	Identifier of the SYNC message

The lower 11bit of the 32bit value contains the identifier (0x80 = 128dez).

Bit 30 = 0: Slave works as Sync Consumer (0x00000080)

Bit 30 = 1: Slave works as Sync Producer (0x40000080)

Object directory

**SYNC interval**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1006	0	Communication cycle period	Unsigned32	rw	N	0x00000000	Maximum length of the SYNC interval in $\mu$ s.

From firmware version V126 and up the Sync-Cycle time can be monitored. Here the following preconditions apply:

- SYNC identifier (Index 0x1005)
  - Bit 30 = 0:lave works as Sync Consumer (0x00000080)
- SYNC interval (Index 0x1006)
  - *Sync-Cycle Time* > 0

Index	Subindex	Value	Default value
0x1019	2	0	0x00000001

The *Transmission-Type* of at least 1 PDO must be set to operation mode 'Sync'.

Mode *Sync Producer* (Index 0x1005):

Sync-Interval is set. There is no monitoring.

**Synchronous Window Length**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1007	0	Synchronous window length	Unsigned32	rw	N	0x00000000	Contains the length of time window for synchronous PDOs in $\mu$ s.

This entry is not evaluated from the 053-1CA00.

**Device name**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1008	0	Manufacturer device name	Visible string	ro	N		Device name of the bus coupler

IM 053-1CA00 = ... CANopen slave IM 053-1CA00.

Since the returned value is longer than 4byte, the segmented SDO protocol is used for transmission.

**Hardware version**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1009	0	Manufacturer Hardware version	Visible string	ro	N		Hardware version number of bus coupler

IM 053-1CA00 = 1.00

Since the returned value is longer than 4byte, the segmented SDO protocol is used for transmission.

**Software version**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100A	0	Manufacturer Software version	Visible string	ro	N		Software version number CANopen software

IM 053-1CA00 = 1.xx

Since the returned value is longer than 4byte, the segmented SDO protocol is used for transmission.

**Node number**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100B	0	Node ID	Unsigned32	ro	N	0x00000000	Node number

The node number is supported for reasons of compatibility.

**Guard time**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100C	0	Guard time[ms]	Unsigned16	rw	N	0x0000	Interval between two guard telegrams. Is set by the NMT master or configuration tool.

**Life time factor**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100D	0	Life time factor	Unsigned8	rw	N	0x00	Life time factor x guard time = life time (watchdog for lifeguarding)

If a guarding telegram is not received within the life time, the node enters the error state. If the life time factor and/or guard time =0, the node does not carry out any life guarding, but can itself be monitored by the master (node guarding).

**Guarding identifier**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x100E	0	COB-ID Guarding Protocol	Unsigned32	ro	N	0x000007xy,xy = node ID	Identifier of the guarding protocol

**Save parameters**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1010	0	Store Parameter	Unsigned8	ro	N	0x01	Number of store Options
	1	Store all parameters	Unsigned32	ro	rw	0x01	Stores all (storable) Parameters

By writing the string "save" in ASCII code (hex code: 0x65766173) into sub-index 1, the current parameters are placed into non-volatile storage (byte sequence at the bus incl. SDO protocol: 0x23 0x10 0x10 0x01 0x73 0x61 0x76 0x65).

If successful, the storage process is confirmed by the corresponding TxSDO (0x60 in the first byte).



*For the bus coupler is not able to send or receive CAN telegrams during the storage procedure, storage is only possible when the node is in pre-operational state.*

*It is recommended to set the complete net to the pre-operational state before storing data to avoid a buffer overrun.*

**Load default values**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1011	0	Restore parameters	Unsigned8	ro	N	0x01	Number of reset options
	1	Restore all parameters	Unsigned32	rw	N	0x01	Resets all parameters to their default values

By writing the string "load" in ASCII code (hex code: 0x6461666C) into sub-index 1, all parameters are set back to default values (delivery state) **at next start-up (reset)** (byte sequence at the bus incl. SDO protocol: 0x23 0x11 0x10 0x01 0x6C 0x6F 0x61 0x64).

This activates the default identifiers for the PDOs.

**Emergency COB-ID**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1014	0	COB-ID Emergency	Unsigned32	ro	N	0x00000080 + Node_ID	Identifier of the emergency telegram

**Consumer Heartbeat Time**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1016	0	Consumer heartbeat time	Unsigned8	ro	N	0x05	Number of entries
	1		Unsigned32	rw	N	0x00000000	Consumer heartbeat time 1
	2		Unsigned32	rw	N	0x00000000	Consumer heartbeat time 2
	3		Unsigned32	rw	N	0x00000000	Consumer heartbeat time 3
	4		Unsigned32	rw	N	0x00000000	Consumer heartbeat time 4
	5		Unsigned32	rw	N	0x00000000	Consumer heartbeat time 5

Structure of the "Consumer heartbeat time" entry:

Bits	31-24	23-16	15-0
Value	Reserved	Node-ID	Heartbeat time
Encoded as	Unsigned8	Unsigned8	Unsigned16

As soon as you try to configure a consumer heartbeat time unequal zero for the same node-ID, the node interrupts the SDO download and throws the error code 0604 0043h.

### Producer Heartbeat Time

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1017	0	Producer heart-beat time	Unsigned16	rw	N	0x0000	Defines the cycle time of heartbeat in ms

### Identity Object

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1018	0	Identity Object	Unsigned8	ro	N	0x04	Contains general information about the device (number of entries)
	1	Vendor ID	Unsigned32	ro	N	<b>0xAFFFAFFE</b>	Vendor ID
	2	Product Code	Unsigned32	ro	N	*	Product Code
	3	Revision Number	Unsigned32	ro	N		Revision Number
	4	Serial Number	Unsigned32	ro	N		Serial Number

\*) Default value Product Code 053-1CA00: 0x0531CA00

### Verify Configuration

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1020	0	Verify Configuration	Unsigned8	ro	N	0x02	Contains general information about the device (number of entries)
	1	Configuration date	Unsigned32	rw	N	0	Date
	2	Configuration time	Unsigned32	rw	N	0	Time

### Modular Devices

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1027	0	Number of connected modules	Unsigned8	ro	N		Contains general information about the device (number of entries)
	1	Module 1	Unsigned16	ro	N		Identification number of Module 1
	...	...	...	...	...	...	...
	N	Module N	Unsigned16	ro	N		Identification number of Module N

The *Identification number* corresponds to the first 4 digits of the *module ID*. The *module ID* may be found at the technical data of the corresponding SLIO module.

**Error Behaviour**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1029	0	Error behaviour	Unsigned8	ro	N	0x02	Number of Error Classes
	1	Communication Error	Unsigned8	rw	N	0x00	Communication Error
	2	Sync-Cycle Error	Unsigned8	rw	N	0x01	Sync-Cycle Error (see Index 0x1006)

As soon as a device failure is detected in "operational" state, the module should automatically change into the "pre-operational" state.

If e.g. an "Error behaviour" is implemented, the module may be configured that its going into STOP at errors.

The following error classes may be monitored:

- 0 = pre-operational
- 1 = no state change
- 2 = stopped

**Communication parameter RxPDO1**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1400	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x8000200 + NODE_ID	COB-ID RxPDO1
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

- Subindex 1 (COB-ID): The lower 11bit of the 32bit value (bits 0-10) contain the CAN identifier, the MSBit (bit 31) shows if the PDO is active (0) or not (1), bit 30 shows if a RTR access to this PDO is permitted (0) or not (1).
- The subindex 2 contains the transmission type.

**Communication parameter RxPDO2**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1401	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0xC8000300 + NODE_ID	COB-ID RxPDO2
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Communication parameter RxPDO3

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1402	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000400 + NODE_ID	COB-ID RxPDO3
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Communication parameter RxPDO4

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1403	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000500 + NODE_ID	COB-ID RxPDO4
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Communication parameter RxPDO5

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1404	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPDO5
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Communication parameter RxPDO6

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1405	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPDO6
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Object directory

**Communication parameter  
RxPDO7**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1406	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPDO7
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

**Communication parameter  
RxPDO8**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1407	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPDO8
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

**Communication parameter  
RxPDO9**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1408	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPDO9
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

**Communication parameter  
RxPDO10**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1409	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPDO10
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Communication parameter RxPDO11

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x140A	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD11
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Communication parameter RxPDO12

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x140B	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD12
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Communication parameter RxPDO13

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x140C	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD13
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Communication parameter RxPDO14

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x140D	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD14
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Communication parameter RxPDO15

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x140E	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD15
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Communication parameter RxPDO16

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x140F	0	Number of Elements	Unsigned8	ro	N	0x02	Communication parameter for the first receive PDOs, subindex 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID RxPD16
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

### Mapping RxPDO1

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1600	0	Number of Elements	Unsigned8	rw	N	0x01	Mapping parameter of the first receive PDO; subindex 0: number of mapped objects
	1	1. mapped object	Unsigned32	rw	N	0x62000108	(2 byte index, 1 byte subindex, 1 byte bit-width)
	2	2. mapped object	Unsigned32	rw	N	0x62000208	(2 byte index, 1 byte subindex, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8. mapped	Unsigned32	rw	N	0x62000808	(2 byte index, 1 byte subindex, 1 byte bit-width)

- The 1. receive PDO (RxPDO1) is per default for the digital outputs. Depending on the number of the inserted outputs, the needed length of the PDO is calculated and mapped into the according objects.
- For the digital outputs are organized in bytes, the length of the PDO can be directly seen in sub-index 0. If the mapping is changed, the entry in sub-index 0 has to be adjusted accordingly.

## Mapping RxPDO2

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1601	0	Number of Elements	Unsigned8	rw	N	0x01	Mapping parameter of the second receive PDO; subindex 0: number of mapped objects
	1	1. mapped object	Unsigned32	rw	N	0x64110110	(2 byte index, 1 byte subindex, 1 byte bit-width)
	2	2. mapped object	Unsigned32	rw	N	0x64110210	(2 byte index, 1 byte subindex, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8. mapped	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte subindex, 1 byte bit-width)

The 2. receive PDO (RxPDO2) is per default for the analog outputs. Depending on the number of the inserted outputs, the needed length of the PDO is calculated and the according objects are mapped. For the digital outputs are organized in words, the length of the PDO can be directly seen in sub-index 0. If the mapping is changed, the entry in sub-index 0 has to be adjusted accordingly.

Mapping RxPDO3-  
RxPDO16

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1602-0 x160F	0	Number of Elements	Unsigned8	rw	N	0x01	Mapping parameter of the 3. to 10. receive PDO; subindex 0: number of mapped objects
	1	1. mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte subindex, 1 byte bit-width)
	2	2. mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte subindex, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8. mapped	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte subindex, 1 byte bit-width)

The receive PDOs 3 to 16 (RxPDO3-16) get an automatic default mapping via the coupler depending from the connected terminals. ↪ *Chap. 4.7 'PDO' page 67*

Communication parameter  
TxPDO1

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1800	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter of the first transmit PDO, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000180 + NODE_ID	COB-ID TxPDO1

## Object directory

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

Sub-index 1 (COB-ID): The lower 11bit of the 32bit value (bits 0-10) contain the CAN identifier, the MSBit (bit 31) shows if the PDO is active (0) or not (1), bit 30 shows if a RTR access to this PDO is permitted (0) or not (1). The sub-index 2 contains the transmission type, sub-index 3 the repetition delay time between two equal PDOs. If an event timer exists with a value unequal 0, the PDO is transmitted when the timer exceeds. If a inhibit timer exists, the event is delayed for this time.

### Communication parameter TxPDO2

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1801	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter of the second transmit PDO, sub-index 0: number of following parameters
	1	COB-ID	Unsigned32	rw	N	0x80000280 + NODE_ID	COB-ID TxPDO2
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO3

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1802	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 3. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000380 + NODE_ID	COB-ID TxPDO3
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO4

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1803	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 4. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000480 + NODE_ID	COB-ID TxPDO4
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

**Communication parameter****TxPDO5**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1804	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 5. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO5
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

**Communication parameter****TxPDO6**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1805	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 6. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO6
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

**Communication parameter****TxPDO7**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1806	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 7. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO7
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO8

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1807	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 8. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO8
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO9

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1808	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 9. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO9
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO10

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1809	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 10. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO10
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO11

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x180A	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 11. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO11
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO12

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x180B	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 12. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO12
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO13

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x180C	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 13. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO13
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO14

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x180D	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 14. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO14
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO15

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x180E	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 15. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO15
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Communication parameter TxPDO16

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x180F	0	Number of Elements	Unsigned8	ro	N	0x05	Communication parameter for the 16. transmit PDO.
	1	COB-ID	Unsigned32	rw	N	0x80000000	COB-ID TxPDO16
	2	Transmission type	Unsigned8	rw	N	0xFF	Transmission type of the PDO
	3	Inhibit time	Unsigned16	rw	N	0x0000	Repetition delay [value x 100 µs]
	5	Event time	Unsigned16	rw	N	0x0000	Event timer [value x 1 ms]

### Mapping TxPDO1

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1A00	0	Number of Elements	Unsigned8	rw	N	depending on the components fitted	Mapping parameter of the first transmit PDO; subindex 0: number of mapped objects
	1	1. mapped object	Unsigned32	rw	N	0x60000108	(2 byte index, 1 byte subindex, 1 byte bit-width)
	2	2. mapped object	Unsigned32	rw	N	0x60000208	(2 byte index, 1 byte subindex, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8. mapped object	Unsigned32	rw	N	0x60000808	(2 byte index, 1 byte subindex, 1 byte bit-width)

The 1. send PDO (TxPDO1) is per default for digital inputs. Depending on the number of the inserted inputs, the needed length of the PDO is calculated and the according objects are mapped. For the digital inputs are organized in bytes, the length of the PDO can be directly seen in sub-index 0. If the mapping is changed, the entry in sub-index 0 has to be adjusted accordingly.

### Mapping TxPDO2

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1A01	0	Number of Elements	Unsigned8	rw	N	depending on the components fitted	Mapping parameter of the second transmit PDO; subindex 0: number of mapped objects
	1	1. mapped object	Unsigned32	rw	N	0x64010110	(2 byte index, 1 byte subindex, 1 byte bit-width)
	2	2. mapped object	Unsigned32	rw	N	0x64010210	(2 byte index, 1 byte subindex, 1 byte bit-width)

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
...	...	...	...	...	...	...	...
	8	8. mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte subindex, 1 byte bit-width)

The 2. send PDO (TxPDO2) is per default for the analog inputs. Depending on the number of the inserted outputs, the needed length of the PDO is calculated and the according objects are mapped. For the digital outputs are organized in words, the length of the PDO can be directly seen in sub-index 0. If the mapping is changed, the entry in sub-index 0 has to be adjusted accordingly.

### Mapping TxPDO3-TxPDO16

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x1A02-0x1A0F	0	Number of Elements	Unsigned8	rw	N	depending on the components fitted	Mapping parameter of the 3. to 10th transmit PDO; subindex 0: number of mapped objects
	1	1. mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte subindex, 1 byte bit-width)
	2	2. mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte subindex, 1 byte bit-width)
	...	...	...	...	...	...	...
	8	8. mapped object	Unsigned32	rw	N	0x00000000	(2 byte index, 1 byte subindex, 1 byte bit-width)

The send PDOs 3 to 16 (TxPDO3-16) get an automatic default mapping via the IM 053-1CA00 depending from the connected terminals. ↪ *Chap. 4.7 'PDO' page 67*

### CAN-Übertragungsrate

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2001	0	CAN transfer rate	Unsigned8	rw	N	0x01	Setting CAN transfer rate

This index entry writes a new transfer rate into the EEPROM. At the next start-up (reset) the CAN coupler starts with the new transfer rate.

Value	CAN transfer rate
"00"	1Mbit/s
"01"	500kbit/s
"02"	250kbit/s
"03"	125kbit/s
"04"	100kbit/s
"05"	50kbit/s
"06"	20kbit/s

Object directory

Value	CAN transfer rate
"07"	10kbit/s
"08"	800kbit/s

### Sync-Cycle Jitter (ms)

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2006	0	Sync-Cycle Jitter (ms)	Unsigned8	rw	N	0x00	Increase Sync-Interval control

The coupler works as *Sync Consumer* (Index: 0x1005) and the sync-cycle time (Index: 0x1006) is to be monitored, so with this object the Sync-Cycle time be increased in order to compensate for any jitter from the sync signal.

### Software package version

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x200A	0	Software package version	Visible string	ro	N		Software package version

IM 053-1CA00 = 1.0.2

Since the returned value is longer than 4byte, the segmented SDO protocol is used for transmission

### SLIO bus version

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x200B	0	SLIO bus version	Unsigned8	ro	N	0x01	SLIO bus version

Possible values:

0x01: SLIO system version 1: Interrupts are not supported

0x02: SLIO system version 2: Interrupts are supported

### Save/Resore status

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2010	0	Save/Restore status	Unsigned32	ro	N	0x65766173 0x64616F6C	All settings stored! Factory settings!

Index 0x2010 shows the save/restore status of the device.

- 0x65766173 (save):  
The current configuration was fix stored by Index 0x1010, subindex 1
- 0x64616F6C (load):  
The device has factory settings .

**Module device name**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2028	0	Number of connected modules	Unsigned8	ro	N		Contains general information about the device (number of entries)
	1	Module 1	Visible string	ro	N		Device name of Module 1
	...	...	...	...	...	...	...
	N	Module N	Visible string	ro	N		Device name of Module N

021-1BF00 = ... 021-1BF00

022-1BF00 = ... 022-1BF00

Since the returned value is longer than 4byte, the segmented SDO protocol is used for transmission.

**Module hardware revision**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2029	0	Number of connected modules	Unsigned8	ro	N		Contains general information about the device (number of entries)
	1	Module 1	Visible string	ro	N		Hardware revision of Module 1 <sup>1</sup>
	...	...	...	...	...	...	...
	N	Module N	Visible string	ro	N		Hardware revision of Module N

1) In conjunction with CAN master 342-1CA70, 21x-2CM0x, 208-1CA00 the data appear in the reverse order. Example: V124 ... CAN-Master: 421V

021-1BF00 = V20

022-1BF00 = V20

The segmented SDO protocol is used for transmission.

**Module software revision**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x202A	0	Number of connected modules	Unsigned8	ro	N		Contains general information about the device (number of entries)
	1	Module 1	Visible string	ro	N		Software revision of Module 1 <sup>1</sup>
	...	...	...	...	...	...	...
	N	Module N	Visible string	ro	N		Software revision of Module N

1) In conjunction with CAN-Master 342-1CA70, 21x-2CM0x, 208-1CA00 the data appear in the reverse order. Example: V124 ... CAN-Master: 421V

Not supported by digital and function modules!

031-1BB30 = V124

The segmented SDO protocol is used for transmission.

**Module serial number**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x202B	0	Number of connected modules	Unsigned8	ro	N		Contains general information about the device (number of entries)
	1	Module 1	Unsigned32	ro	N		Serial number of Module 1
	...	...	...	...	...	...	...
	N	Module N	Unsigned32	ro	N		Serial number of Module N

**CAN coupler FPGA version**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2030	0	FPGA version	Unsigned16	ro	N		FPGA Version

**Module FPGA version**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2031	0	Number of connected modules	Unsigned8	ro	N		Contains general information about the device (number of entries)
	1	Module 1	Unsigned16	ro	N		FPGA version of Module 1
	...	...	...	...	...	...	...
	N	Module N	Unsigned16	ro	N		FPGA version of Module N

**CAN coupler MxFile**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2040	0	MxFile	Visible string	ro	N		MxFile

053-1CA00 = MX000029.xyz

The segmented SDO protocol is used for transmission.

**Module MxFile**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2041	0	Number of connected modules	Unsigned8	ro	N		Contains general information about the device (number of entries)
	1	Module 1	Visible string	ro	N		MxFile of Module 1
	...	...	...	...	...	...	...
	N	Module N	Visible string	ro	N		MxFile of Module N

021-1BF00 = MX000006.xyz

022-1BF00 = MX000015.xyz

The segmented SDO protocol is used for transmission.

**CAN coupler Product version**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2050	0	FPGA version	Visible string	ro	N		Product Version

053-1CA00 = 01.V10.001

The segmented SDO protocol is used for transmission.

**Module Product version**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2051	0	Number of connected modules	Unsigned8	ro	N		Contains general information about the device (number of entries)
	1	Module 1	Visible string	ro	N		Product version of Module 1
	...	...	...	...	...	...	...
	N	Module N	Visible string	ro	N		Product version of Module N

**PDO-Control**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2400	0	Number of Elements	Unsigned8	ro	N	0x10	Time control for RxPDOs
	1	RxPDO1	Unsigned16	rw	N	0x0000	Timer value [ms]
	2	RxPDO2	Unsigned16	rw	N	0x0000	Timer value [ms]
	...	...	...	...	...	...	...
	16	RxPDO16	Unsigned16	rw	N	0x0000	Timer value [ms]

The control starts as soon as the timer is unequal 0. Every received RxPDO resets the timer. When the timer has been expired, the CAN coupler switches into the state "pre-operational" and sends an emergency telegram.

**Module Diagnostic data record 0**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2F00	0	Number of connected modules	Unsigned8	ro	N		Contains general information about the device (number of entries)
	1	Module 1	Octet string	ro	N		Diagnostic data record 0 of Module 1
	...	...	...	...	...	...	...
	N	Module N	Octet string	ro	N		Diagnostic data record 0 of Module N

The record set 0 is 4 byte long and corresponds to the first 4 byte of the diagnostics data.

Object directory

**Module Diagnostic data record 1**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2F01	0	Number of connected modules	Unsigned8	ro	N		Contains general information about the device (number of entries)
	1	Module 1	Octet string	ro	N		Diagnostic data record 1 of Module 1
	...	...	...	...	...	...	...
	N	Module N	Octet string	ro	N		Diagnostic data record 1 of Module N

The segmented SDO protocol is used for transmission.

**Module Initialization error**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x2F02	0	Number of connected modules	Unsigned8	ro	N		Contains general information about the device (number of entries)
	1	Module 1	Unsigned16	ro	N		Initialization error of Module 1
	...	...	...	...	...	...	...
	N	Module N	Unsigned16	ro	N		Initialization error of Module N

**Possible values:**

Bit	Error
0	CRC error
1	FPGA version is too old
4	MxFile - error in group 0
5	MxFile - error in group x
11	unknown module
12	reserved
13	Configuration error, i.e. the module configuration was stored with index 0x1010, after a start-up the module configuration was changed.
14	There are too many modules of the type AI, AO, counter and CP installed.
15	reserved

**0000:** no Module Initialization error

## 1. Module Parameter

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x3100	0	Number of Elements	Unsigned8	ro	N	depending on the number of configurable modules	Number of configurable modules 0x00 : no module available 0xXX : XX number of configurable modules available
	1	1. Prm 1. Module	Unsigned8	wo	N	depending on the components fitted	1. Parameter byte of the 1. configurable module
	2	1. Prm 2. Module	Unsigned8	wo	N	depending on the components fitted	1. Parameter byte of the 2. configurable module
	...	...	...	...	...	...	...
	64	1. Prm 64. Module	Unsigned8	wo	N	depending on the components fitted	1. Parameter byte of the 64. configurable module

The number of configurable modules may be determined with index 0x3100 and subindex 0.

Via subindex 1 the 1. parameter byte of the 1. configurable module may be accessed. Please consider that this value does not correspond to the slot number but to the position within the configurable modules.

The 1. parameter byte of the 1. configurable module is assigned to subindex 1, the 1. parameter byte of the 2. module is assigned to subindex 2, and so on.

## 2. ... 65. Module Parameter

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x3101	1	2. Prm 1. Module	Unsigned8	wo	N	depending on the components fitted	2. Parameter byte of the 1. configurable module
	2	2. Prm 2. Module	Unsigned8	wo	N	depending on the components fitted	2. Parameter byte of the 2. configurable module
	...	...	...	...	...	...	...
	64	2. Prm 64. Module	Unsigned8	wo	N	depending on the components fitted	2. Parameter byte of the 64. configurable module
0x3102	1	3. Prm 1. Module	Unsigned8	wo	N	depending on the components fitted	3. Parameter byte of the 1. configurable module
	2	3. Prm 2. Module	Unsigned8	wo	N	depending on the components fitted	3. Parameter byte of the 2. configurable module
	...	...	...	...	...	...	...
	64	3. Prm 64. Module	Unsigned8	wo	N	depending on the components fitted	3. Parameter byte of the 64. configurable module
...	...	...	...	...	...	...	...

## Object directory

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x3140	1	65. Prm 1. Module	Unsigned8	wo	N	depending on the components fitted	65. Parameter byte of the 1. configurable module
	2	65. Prm 2. Module	Unsigned8	wo	N	depending on the components fitted	65. Parameter byte of the 2. configurable module
	...	...	...	...	...	...	...
	64	65. Prm 64. Module	Unsigned8	wo	N	depending on the components fitted	65. Parameter byte of the 64. configurable module

The 2. ... 65. parameters of one module may be accessed via index 0x3101 ... 0x3140.  
Here the 1. ... 64. module may be defined via subindex 1 ... 64.

**Write Parameters**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x31FE	0	Number of Elements	Unsigned8	ro	N	depending on the number of configurable modules	Number of configurable modules 0x00: no module available 0xXX: XX number of configurable modules available
	1	Prm 1. Module	Unsigned8	wo	N	0x00 ... 0xFF	Writes parameters to 1. module with any write access
	2	Prm 2. Module	Unsigned8	wo	N	0x00 ... 0xFF	Writes parameters to 2. module with any write access
	...	...	...	...	...	...	...
	64	Prm 64. Module	Unsigned8	wo	N	0x00 ... 0xFF	Writes parameters to 64. module with any write access

The number of configurable modules may be determined with index 0x31FE and sub-index 0.

The activation of the preset parameters via index 0x3100 ... 0x3140 happens by any write access to index 0x31FE. Here the module 1 to 64 is defined by subindex 1 ... 64.

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x31FF	0	Parameters to all modules	Unsigned8	wo	N	0	Writes parameters to all modules

The parameters may be written to all modules with index 0x31FF and subindex 0.

**Example**

Your System SLIO consists of the following configurable modules:

- 031-1BB30
- 032-1BB30

The parameters may be accessed with the following indexes:

**Change parameters**

Index	Subindex	
	1	2
	031-1BB30	032-1BB30
0x3100	Function channel 0	reserved
0x3101	Function channel 1	Short circuit recognition
0x3102	reserved	Function Channel 0
0x3103	reserved	Function Channel 1

**Write parameters**

Index	Subindex	
	1	2

## Object directory

	031-1BB30	032-1BB30
0x31FE	Write parameters	Write parameters

### Access to record set bus coupler

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x3200	0x00	Number of Elements	Unsigned8	ro	N	0xFF	Number of record sets
	0x50	Device name	Visib. string	ro	N		Device name
	0x51	HW Rev.	Visib. string	ro	N		Hardware revision <sup>1</sup>
	0x52	SW Rev.	Visib. string	ro	N		Software revision <sup>1</sup>
	0x53	Serial No.	Unsigned32	ro	N		Serial number
	0x54	FPGA vers.	Unsigned16	ro	N		FPGA version
	0x55	MX File name	Visib. string	ro	N		MX file name
	0x56	Product version	Visib. string	ro	N		Product version
	0x57	Order code	Visib. string	ro	N		Order code

1) In conjunction with CAN master 342-1CA70, 21x-2CM0x, 208-1CA00 the data appear in the reverse order. Example: V124 ... CAN master: 421V

With index 0x3200 the bus coupler may be accessed. Here the record set may be defined with the corresponding sub-index.

### Access to record set function modules

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x3201 ... 0x3240	0x00	Number of Elements	Unsigned8	ro	N	0xFF	Number of record sets of the modules at slot 1 ... 64
	0x01	Diag RS1	Octet string	r	N		Diagnostics record set 1
	0x02	Diag RS0	Octet string	r	N		Diagnostics record set 0
	0x01	Param RS1	Octet string	w	N		Parameter record set 1
	0x02	Param RS0	Octet string	w	N		Parameter record set 0
	0x50	Device name	Visib. string	ro	N		Device name
	0x51	HW Rev.	Visib. string	ro	N		Hardware revision <sup>1</sup>
	0x52	SW Rev.	Visib. string	ro	N		Software revision <sup>1</sup>
	0x53	Serial No.	Unsigned32	ro	N		Serial number
	0x54	FPGA vers.	Unsigned16	ro	N		FPGA version
	0x55	MX File name	Visib. string	ro	N		MX file name
	0x56	Product version	Visib. string	ro	N		Product version
	0x57	Order code	Visib. string	ro	N		Order code
	0x60	Node err. count	Visib. string	ro	N		Node error counter
	0x61	Node err. count + Reset	Visib. string	ro	N		Node error counter and set counter to 0

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
	0x7D	RS 0...N	Octet string	rw	N		Record set 0...N
	0x7E	RS 0	Octet string	rw	N		Record set 0
	0x7F	RS 1	Octet string	rw	N		Record set 1
	0x80	RS 128	Octet string	rw	N		Record set 128
	0x81	RS 129	Octet string	rw	N		Record set 129
	...	...	...	...	...		...
	0xAF	RS 175	Octet string	rw	N		Record set 175

1) In conjunction with CAN master 342-1CA70, 21x-2CM0x, 208-1CA00 the data appear in the reverse order. Example: V124 ... CAN master: 421V

The record sets of the assigned function modules at slot 1 ... 64 may be accessed by the index 0x3201 ... 0x3240. Here the record set may be defined with the corresponding sub-index.



*Please regard with a write access to sub-index 0x01/0x02 the record set 1/0 of the parameter data is written but with a read access the record set 1/0 of the diagnostics data is read!*

### 2/4bit Digital inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5000	0x00	2/4bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1. input block	Unsigned8	ro	Y		1. digital input block
	...	...	...	...	...	...	...
	0x40	64. input block	Unsigned8	ro	Y		64. digital input block

### 2/4bit Polarity Digital inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5002	0x00	2/4bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1. input block	Unsigned8	rw	N	0x00	1. polarity digital input block
	...	...	...	...	...	...	...
	0x40	64. input block	Unsigned8	rw	N	0x00	64. polarity digital input block

Individual inversion of the input channels

1 = Input inverted

0 = Input not inverted

Object directory

**2/4bit Digital outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5200	0x00	2/4bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	Y		1. digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	Y		64. digital output block

**2/4bit Change Polarity Digital outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5202	0x00	2/4bit digital output block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	N	0x00	1. polarity digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	N	0x00	64. polarity digital output block

Individual inversion of the output channels

1 = Output inverted

0 = Output not inverted

**2/4bit Error Mode Digital outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5206	0x00	2/4bit digital output block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	N	0xFF	1. error mode digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	N	0xFF	64. error mode digital output block

By means of this object you may define whether in an event of an error an output channel takes a defined value, which may be defined with object 0x5207.

1 = take value defined at object 0x5207

0 = on error fix output value

**2/4bit Error Value Digital outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5207	0x00	2/4bit digital output block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	N	0x00	1. error value digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	N	0x00	64. error value digital output block

With an activated error mode, in an event of an error the value preset here is taken.

**Counter value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5400	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available counter values
	0x01	1. counter value	Unsigned32	ro	Y		1. counter value
	...	...	...	...	...	...	...
	0x20	32. counter value	Unsigned32	ro	Y		32. counter value

**Counter latch value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5401	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available counter latch values
	0x01	1. counter latch value	Unsigned32	ro	Y		1. counter latch value
	...	...	...	...	...	...	...
	0x20	32. counter latch value	Unsigned32	ro	Y		32. counter latch value

**Counter status value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5402	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available counter status values
	0x01	1. counter status value	Unsigned16	ro	Y		1. counter status value
	...	...	...	...	...	...	...
	0x20	32. counter status value	Unsigned16	ro	Y		32. counter status value

Object directory

**Counter  $\mu$ s Ticker value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5403	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available counter $\mu$ s ticker values
	0x01	1. counter $\mu$ s ticker value	Unsigned16	ro	Y		1. counter $\mu$ s ticker value
	...	...	...	...	...	...	...
	0x20	32. counter $\mu$ s ticker value	Unsigned16	ro	Y		32. counter $\mu$ s ticker value

**SSI value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5410	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available SSI values
	0x01	1. SSI value	Unsigned32	ro	Y		1. SSI value
	...	...	...	...	...	...	...
	0x10	16. SSI value	Unsigned32	ro	Y		16. SSI value

**SSI  $\mu$ sTicker value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5411	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available SSI $\mu$ s ticker values
	0x01	1. SSI $\mu$ s ticker value	Unsigned16	ro	Y		1. SSI $\mu$ s ticker value
	...	...	...	...	...	...	...
	0x10	16. SSI $\mu$ s ticker value	Unsigned16	ro	Y		16. SSI $\mu$ s ticker value

**PWM status value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5420	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available PWM status values
	0x01	1. PWM status value	Unsigned16	ro	Y		1. PWM status value
	...	...	...	...	...	...	...
	0x10	16. PWM status value	Unsigned16	ro	Y		16. PWM status value

**ETS Digital inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5430	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available ETS input values
	0x01	1. ETS value	Unsigned32	ro	Y		1. ETS value
	...	...	...	...	...	...	...
	0x3C	60. ETS value	Unsigned32	ro	Y		60. ETS value

**µs ticker time**

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x5431	0	µs ticker[ms]	Unsigned32	ro	Y	0x0000	Time of the 32bit System SLIO µs ticker.

**Status ETS Digital outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5440	0x00	Number of modules	Unsigned8	ro	N	depending on the components fitted	Number of available ETS output modules
	0x01	1. ETS module	Unsigned32	ro	Y		Status of 1. ETS out module
	...	...	...	...	...	...	...
	0x04	4. ETS module	Unsigned32	ro	Y		Status of 4. ETS out module

**Counter compare value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5600	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available counter compare values
	0x01	1. counter compare value	Unsigned32	rw	Y		1. counter compare value
	...	...	...	...	...	...	...
	0x20	32. counter compare value	Unsigned32	rw	Y		32. counter compare value

Object directory

**Counter set value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5601	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available counter set values
	0x01	1. counter set value	Unsigned32	rw	Y		1. counter set value
	...	...	...	...	...	...	...
	0x20	32. counter set value	Unsigned32	rw	Y		32. counter set value

**Counter control value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5602	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available counter control values
	0x01	1. counter control value	Unsigned16	rw	Y		1. counter control value
	...	...	...	...	...	...	...
	0x20	32. counter control value	Unsigned16	rw	Y		32. counter control value

**PWM Pulse duration value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5620	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available PWM pulse duration values
	0x01	1. PWM value	Unsigned32	rw	Y		1. PWM pulse duration value
	...	...	...	...	...	...	...
	0x10	16. PWM value	Unsigned32	rw	Y		16. PWM pulse duration value

**PWM control value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5621	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available PWM control values
	0x01	1. PWM control value	Unsigned16	rw	Y		1. PWM control value
	...	...	...	...	...	...	...
	0x10	16. PWM control value	Unsigned16	rw	Y		16. PWM control value

**ETS Digital outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x5640	0x00	Number of max. entries	Unsigned8	ro	N	depending on the components fitted	Number of available ETS output values
	0x01	1. ETS value	Unsigned32	rw	Y		1. ETS value
	...	...	...	...	...	...	...
	0x3C	60. ETS value	Unsigned32	rw	Y		60. ETS value

**8bit Digital inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6000	0x00	8bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1. input block	Unsigned8	ro	Y		1. digital input block
	...	...	...	...	...	...	...
	0x40	64. input block	Unsigned8	ro	Y		64. digital input block

**8bit Polarity Digital inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6002	0x00	8bit digital input block	Unsigned8	ro	N	0x01	Number of available digital 8bit input blocks
	0x01	1. input block	Unsigned8	rw	N	0x00	1. polarity digital input block
	...	...	...	...	...	...	...
	0x40	64. input block	Unsigned8	rw	N	0x00	64. polarity digital input block

Individual inverting of input polarity

1 = input inverted

1 = input inverted

**8bit Digital outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6200	0x00	8bit digital output block	Unsigned8	ro	N	0x01	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	Y		1. digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	Y		64. digital output block

**8bit Change Polarity Digital outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6202	0x00	8bit digital output block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	N	0x00	1. polarity digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	N	0x00	64. polarity digital output block

Individual inverting of input channels:

- 1 = input inverted
- 0 = input not inverted

**8bit Error Mode Digital outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6206	0x00	8bit digital output block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	N	0xFF	1. error mode digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	N	0xFF	64. error mode digital output block

This object indicates whether an output is set to a pre-defined error value (set in object 0x6207) in case of an internal device failure.

- 1 = overtake the value from object 0x6207
- 0 = keep output value in case of error

**8bit Error Value Digital outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6207	0x00	8bit digital output block	Unsigned8	ro	N	Depending on the components fitted	Number of available digital 8bit output blocks
	0x01	1. output block	Unsigned8	rw	N	0x00	1. error value digital output block
	...	...	...	...	...	...	...
	0x40	64. output block	Unsigned8	rw	N	0x00	64. error value digital output block

Presupposed that the error mode is active, device failures set the output to the value configured by this object.

**Analog inputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6401	0x00	2byte input block	Unsigned8	ro	N	depending on the components fitted	Number of available analog inputs
	0x01	1. input channel	Unsigned16	ro	Y		1. analog input channel
	...	...	...	...	...	...	...
	0xFE	254. input channel	Unsigned16	ro	Y		254. analog input channel

**Analog outputs**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6411	0x00	2byte output block	Unsigned8	ro	N	depending on the components fitted	Number of available analog outputs
	0x01	1. output channel	Unsigned16	rw	Y		1. analog output channel
	...	...	...	...	...	...	...
	0xFE	254. output channel	Unsigned16	rw	Y		254. analog output channel

**Analog Input Interrupt  
Trigger selection**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6421	0x00	Number of Inputs	Unsigned8	ro	N	depending on the components fitted	Number of available analog inputs
	0x01	Trigger 1. input channel	Unsigned8	rw	N	0x07	Input interrupt trigger for 1. analog input channel
	...	...	...	...	...	...	...
	0xFE	Trigger 254. input channel	Unsigned8	rw	N	0x07	Input interrupt trigger for 254. analog input channel

This object determines which events shall cause an interrupt for a specific channel. Bits set in the list below refer to the interrupt trigger.

Bit no.	Interrupt trigger
0	Upper limit exceeded 6424
1	Input below lower limit 6425
2	Input changed by more than negative delta 6426
3 to 7	Reserved

### Analog Input Interrupt Source

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6422	0x00	Number of Interrupt	Unsigned8	ro	N	0x01	Number of interrupt source bank
	0x01	Interrupt source bank	Unsigned32	ro	N	0x00000000	Interrupt source bank 1
	0x02	Interrupt source bank	Unsigned32	ro	N	0x00000000	Interrupt source bank 2

This object defines the channel that is responsible for the Interrupt. Bits set refer to the number of the channel that caused the Interrupt. The bits are automatically reset, after they have been read by a SDO or send by a PDO.

- 1 = Interrupt produced
- 0 = Interrupt not produced

### Event driven analog inputs

Index	Sub-index	Name	Type	Attr.	Map.	Default value	Meaning
0x6423	0x00	Global interrupt enable	Boolean	rw	N	FALSE ("0")	Activates the event-driven transmission of PDOs with analog inputs

Although the analog inputs are -acc. to CANopen - per default set to the transmission type 255 (event triggered) in the TxPDO2, the "event" (the alteration of an input value) is suppressed by the event control in object 0x6423 in order to prevent the bus from being swamped with analog signals. Before activation, it is convenient to parameterize the transmission behavior of the analog PDOs:

- inhibit time (object 0x1800ff, sub-index 3)
- limit value monitoring (objects 0x6424 + 0x6425)
- delta function (object 0x6426)

### Upper limit value analog inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6424	0x00	Number of Inputs	Unsigned8	ro	N	depending on the components fitted	Number of available analog inputs
	0x01	Upper limit 1. input channel	Unsigned32	rw	N	0x00000000	Upper limit value for 1. analog input channel
	...	...	...	...	...	...	...
	0xFE	Upper limit 254. input channel	Unsigned32	rw	N	0x00000000	Upper limit value for 254. analog input channel

Values unequal to zero are activating the upper limit value for this channel. A PDO is then transmitted when the upper limit value is exceeded. In addition, the event trigger has to be active (object 0x6423). The data format corresponds to that of the analog inputs.

### Lower limit value analog inputs

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6425	0x00	Number of Inputs	Unsigned8	ro	N	depending on the components fitted	Number of available analog inputs
	0x01	Lower limit 1. input channel	Unsigned32	rw	N	0x00000000	Lower limit value for 1. analog input channel
	...	...	...	...	...	...	...
	0xFE	Lower limit 254. input channel	Unsigned32	rw	N	0x00000000	Lower limit value for 254. analog input channel

Values unequal to zero are activating the lower limit value for this channel. A PDO is then transmitted when the lower limit value is underrun. In addition, the event trigger has to be active (object 0x6423). The data format corresponds to that of the analog inputs.

### Delta function

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6426	0x00	Number of Inputs	Unsigned8	ro	N	depending on the components fitted	Number of available analog inputs
	0x01	Delta value 1. input channel	Unsigned32	rw	N	0x00000002	Delta value for 1. analog input channel
	...	...	...	...	...	...	...
	0xFE	Delta value 254. input channel	Unsigned32	rw	N	0x00000002	Delta value for 254. analog input channel

Values unequal to zero are activating the delta function for this channel. A PDO is then transmitted when the value has been changed for more than the delta value since the last transmission. In addition, the event trigger has to be active (object 0x6423). The data format corresponds to that of the analog inputs. (The delta function accepts only positive values.)

### Analog Output Error Mode

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6443	0x00	Analog output block	Unsigned8	ro	N	Depending on the components fitted	Number of available analog outputs
	0x01	1. analog output block	Unsigned8	rw	N	0xFF	1. error mode analog output block
	...	...	...	...	...	...	...
	0xFE	254. analog output block	Unsigned8	rw	N	0xFF	254. error mode analog output block

This object indicates whether an output is set to a pre-defined error value (set in object 0x6444) in case of an internal device failure.

0 = current value

1 = set to error value 0x6444

**Analog Output Error Value**

Index	Sub-Index	Name	Type	Attr.	Map.	Default value	Meaning
0x6444	0x00	16bit digital input block	Unsigned8	ro	N	Depending on the components fitted	Number of available analog output blocks
	0x01	1. analog block	Unsigned16	rw	N	0x0000	1. analog output block
	...	...	...	...	...	...	...
	0xFE	254. analog block	Unsigned16	rw	N	0x0000	254. analog output block

Presupposed that the corresponding error (0x6443) is active, device failures set the output to the value configured by this object.

<b>SDO Abort Codes</b>	0x05030000 //Toggle bit not alternated
	0x05040000 //SDO protocol timed out
	0x05040001 //Client/server command specify not valid or unknown
	0x05040002 //Invalid block size (block mode only)
	0x05040003 //Invalid sequence number (block mode only)
	0x05040004 //CRC error (block mode only)
	0x05040005 //Out of memory
	0x06010000 //Unsupported access to an object
	0x06010001 //Attempt to read a write only object
	0x06010002 //Attempt to write a read only object
	0x06020000 //Object does not exist in the object dictionary
	0x06040041 //Object cannot be mapped to the PDO
	0x06040042 //The number and length of the objects to be mapped would exceed PDO length
	0x06040043 //General parameter incompatibility reason
	0x06040047 //General internal incompatibility in the device
	0x06060000 //Access failed due to an hardware error
	0x06070010 //Data type does not match, length of service parameter does not match
	0x06070012 //Data type does not match, length of service parameter too high
	0x06070013 //Data type does not match, length of service parameter too low
	0x06090011 //Sub-index does not exist
	0x06090030 //Value range of parameter exceeded (only for write access)
	0x06090031 //Value of parameter written too high
	0x06090032 //Value of parameter written too low
	0x06090036 //Maximum value is less than minimum value
	0x08000000 //general error
	0x08000020 //Data cannot be transferred or stored to the application
	0x08000021 //Data cannot be transferred or stored to the application because of local control
	0x08000022 //Data cannot be transferred or stored to the application because of the present device state
	0x08000023 //Object dictionary dynamic generation fails or no object dictionary is present (e.g. object dictionary is generated from file and generation fails because of an file error)

## 4.10 Emergency Object

### Overview

The System SLIO IM 053-1CA00 is provided with the emergency object to notify other devices connected to the CANopen bus about internal error events or CAN-Bus errors. It has a high priority and gives you important information about the states of device and network.



*We strongly recommend to evaluate the emergence object - it is an important information pool!*

### Telegram structure

The emergency telegram has always a length of 8byte. It starts with 2byte error code followed by the 1byte error register and closes with 5byte additional code.

Error code (Low Byte)	Error code (High Byte)	Error Register Index 0x1001	Info 0	Info 1	Info 2	Info 3	Info 4
--------------------------	---------------------------	--------------------------------	--------	--------	--------	--------	--------

### Error messages

Error Code	Meaning	Info 0	Info 1	Info 2	Info 3	Info 4	Add. Inf. (H Byte)	Add. Inf. (L Byte)
0x0000	Reset Emergency							
0x0000	Process alarm	0x80+Slot No.	Diag. Byte 1	Diag. Byte 2	Diag. Byte 3	Diag. Byte 4	0x80	Slot No.
0x1000	Module configuration has changed and Index 0x1010 is equal to "save"	0x01	0x00	0x00	0x00	0x00		
0x1000	Diagnostic alarm	0x40+Slot No.	Diag. Byte 1	Diag. Byte 2	Diag. Byte 3	Diag. Byte 4	0x40	Slot No.
0x1000	Initialization error on System SLIO bus	0xE3	Slot No.	Index 0x2F02 L Byte Err. Bits	Index 0x2F02 H Byte Err. Bits	0x00	0xE3	Slot No.
0x1000	Initialization error on System SLIO bus during initialization	0xE0	0x00	0x00	0x00	0x00	0xE0	0x00
0x1000	Error on System SLIO bus	0xE1	0x00	0x00	0x00	0x00	0xE1	0x00
0x2000	IO size CP module > 8byte	0xE2	0x00	0x00	0x00	0x00	0xE2	0x00
0x3100	Main voltage error	0x00	0x00	0x00	0x00	0x00	---	---
0x8101	Sync-Cycle Error	Cycle Time 0 ... 7	Cycle Time 8 ... 15	Cycle Time 16 ... 23	Cycle Time 24 ... 31	0x00	---	---
0x8130	Heartbeat Consumer	Index (1 ... 5)	Node ID	L Byte Timer Value	H Byte Timer Value	0x00	---	---
0x8157	System SLIO bus error	0x05	Slot No.	0x00	0x00	0x00	Slot No.	0x00
0x8130	Node Guarding Error	L Byte GuardTime	H Byte GuardTime	LifeTime	0x00	0x00	---	---

Error Code	Meaning	Info 0	Info 1	Info 2	Info 3	Info 4	Add. Inf. (H Byte)	Add. Inf. (L Byte)
0x8210	PDO not processed due to length error	PDO No.	Wrong length	PDO length	0x00	0x00	---	---
0x8220	PDO length exceeded	PDO No.	Wrong length	PDO length	0x00	0x00	---	---

## 4.11 Network management

### 4.11.1 Overview

#### Overview

Network management (NMT) provides the global services specifications for network supervision and management. This includes the login and logout of the different network devices, the supervision of these devices as well as the processing of exceptions.

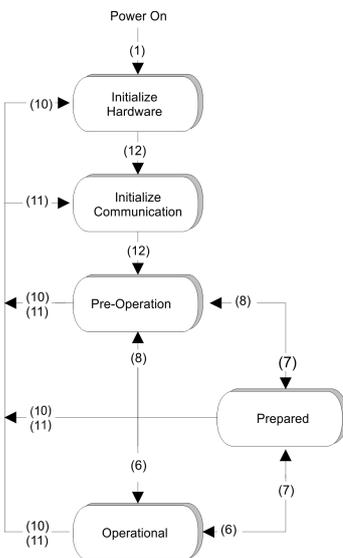
NMT service messages have the COB identifier 0x0000. An additional module-ID is not required. The length is always 2 data bytes.

The 1. data byte contains the NMT command specifier: **CS**.

The 2. data byte contains the module-ID (0x00 for broadcast command).

#### States

The following picture shows an overview over all CANopen status changes and the corresponding NMT command specifiers:



- (1) The initialization state is reached automatically after start-up.
- (6) "Start\_Remote\_Node" (CS: 0x01) Starts the module, releases outputs and starts the PDO transmission.
- (7) "Stop\_Remote\_Node" (CS: 0x02) Outputs are switching into error state, SDO and PDO are switched off.
- (8) "Enter\_Pre-operational\_State" (CS:0x80) Stops PDO transmission, SDO still active.
- (10) "Reset\_Node" (CS:0x81) Executes reset. All objects are set back to PowerOn defaults.
- (11) "Reset\_Communication" (CS:0x82) Executes reset of the communication functions. Objects 0x1000 - 0x1FFF are set back to PowerOn defaults.
- (12) After initialization the state "pre-operational is automatically reached - here the boot-up message is send.

### 4.11.2 Node Guarding

- The bus coupler also supports the Node Guarding object as defined by CANopen to ensure that other devices on the bus are supervised properly.
- Node Guarding operation is started when the first guard requests (RTR) is received from the master. The respective COB identifier is permanently set to 0x700 + module-ID at variable 0x100E in the object directory. If the coupler does not receive a guard request message from the master within the "guard time" (object 0x100C) when the node guarding mode is active the module assumes that the master is not operating properly. When the time determined by the product of "guard time" (0x100C) and "life-time factor" (0x100D) has expired, the module will automatically assume the status "pre-operational".
- When either the "guard time" (object 0x100C) or the "life-time factor" (0x100D) has been set to zero by an SDO download from the master, the expiry of the guard time is not monitored and the module remains in its current operating mode.

### 4.11.3 Heartbeat

- The 053-1CA00 also supports the Heartbeat Mode in addition to Node Guarding.
- When a value is entered into index 0x1017 (Heartbeat Producer Time) then the device status (Operational, Pre-Operational,...) of the bus coupler is transferred by means of the COB identifier (0x700+module-ID) when the heartbeat timer expires.
- The Heartbeat Mode starts automatically as soon as the index 1017h contains a value that is larger than 0.